Visual Basic 6.0
untuk Media Pembelajaran Interaktif

Dr. Efron Manik, M.Si

Visual Basic 6.0
untuk Media Pembelajaran Interaktif

Dr. Efron Manik, M.Si

Penerbit:

LPPM UHN Press

KATA PENGANTAR

Kami mengucapkan terima kasih kepada Tuhan yang maha Esa karena anugrah-Nya penulisan buku ini dapat diselesaikan. Buku ini memuat cocok untuk diperuntukkan bagi orang belajar pemrograman dengan Bahasa Visual Basic 6.0. Penyajiannya sangat sederhana dan memuat banyak contoh proyek pemrograman sehingga dapat digunakan untuk belajar tanpa kehadiran seorang pengajar. Contoh-contohnya banyak berbentuk proyek pembuatan media pembelajaran interaktif sehingga cocok untuk orang yang tertarik dengan pembelajaran era industry 4.0. Kami menyadari bahwa modul ini masih jauh dari sempurna, sehingga kami berharap pembaca dapat memberikan masukan yang membangun demi kemajuan kita semua.

Pada kesempatan ini, kami juga berterimakasih kepada semua rekan-rekan yang terlibat yang sudah memberi masukan untuk perbaikan buku ini. Semoga buku ini akan bermanfaat bagi semua pembaca.

Hormat kami
Penulis

DAFTAR ISI

Bab 1
Pengantar

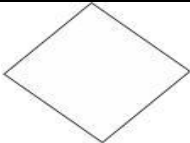
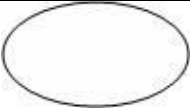
Sebelum membuat program, programmer biasanya terlebih dahulu membuat diagram alur atau algoritma. Diagram alur atau algoritma akan membantu melihat program secara keseluruhan. Jika diagram alur atau algoritma sudah ditulis dengan benar dan rinci maka pekerjaan programmer sudah selesai lebih dari separuh. Diagram alur dan algoritma selalu sama untuk semua bahasa pemrograman, termasuk bahasa Visual Basic. Sejarah Baisc, QBasic, dan Visual Basic akan dibahas terlebih dahulu sebelum belajar pemrograman. Setelah belajar Bab 01, mahasiswa diharapkan mampu menjelaskan sejarah bahasa Visual Basik, serta mampu membuat diagram alur dan algoritma untuk suatu kasus sederhana.

1.1. Diagram Alur

Diagram alur kadang disebut flowchart. Diagram alur merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran proses. Diagram alur menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya

dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

Tabel 1. Simbol Diagram Alur

Gambar	Simbol untuk	Keterangan
	Proses/ Langkah	Menyatakan kegiatan yang akan ditampilkan dalam diagram alir.
	Titik Keputusan	Proses / Langkah dimana perlu adanya keputusan atau adanya kondisi tertentu. Di titik ini selalu ada dua keluaran untuk melanjutkan aliran kondisi yang berbeda.
	Masukan / Keluaran Data	Digunakan untuk mewakili data masuk, atau data keluar.
	Terminasi	Menunjukkan awal atau akhir sebuah proses.
	Garis alir	Menunjukkan arah aliran proses atau algoritma.
	Kontrol / Inspeksi	Menunjukkan proses / langkah dimana ada inspeksi atau pengontrolan.

Tabel 1 adalah simbol flowchart yang umum digunakan. Untuk membuat diagram alur. Perhatikan Contoh 1 berikut!

Contoh 1

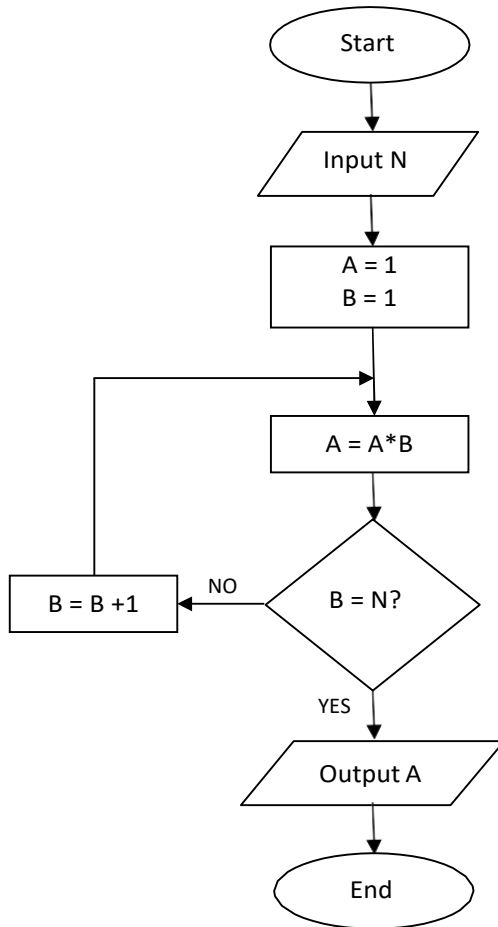



Diagram di atas merupakan diagram alur untuk menghitung factorial N (N!). Setelah dimulai (Start) program meminta masukan bilangan N (Input N). Setelah itu program langsung memberi nilai awal untuk variabel A = 1 dan B = 1. Selanjutnya program mengerjakan bahwa nilai A yang baru sama dengan nilai A sebelumnya dikali B. Kemudian diuji kesamaan dua nilai yaitu: apakah sama nilai B dan N?. Jika tidak (NO) maka dikerjakan pekerjaan bahwa nilai B yang baru sama dengan nilai B sebelumnya ditambah satu dan selanjutnya kembali kepada pekerjaan A = A*B. Tetapi jika ya (YES) maka hasilnya dicetak nilai A (Output A). Akhirnya program berakhir (End).

1.2. Algoritma

Dalam matematika dan komputasi, algoritma merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, dalam hal ini berbeda dengan heuristik. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai. Diagram Alur sering digunakan untuk menggambarkan sebuah algoritma.

Desain dan analisis algoritma adalah suatu cabang khusus dalam ilmu komputer yang mempelajari karakteristik dan performa dari

suatu algoritma dalam menyelesaikan masalah, terlepas dari implementasi algoritma tersebut. Dalam cabang disiplin ini algoritma dipelajari secara abstrak, terlepas dari sistem komputer atau bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan pada suatu masalah dengan kriteria yang sama.

Kompleksitas dari suatu algoritma merupakan ukuran seberapa banyak komputasi yang dibutuhkan algoritma tersebut untuk menyelesaikan masalah. Secara informal, algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki kompleksitas yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan masalahnya mempunyai kompleksitas yang tinggi.

Contoh 2

Perhatikan Diagram Alur menghitung faktorial pada Contoh 1. Kita akan menuliskan algoritma dari contoh tersebut. Bentuk umumnya terdiri dari input, output, dan langkah-langkah.

Algoritma Menghitung Faktorial

Input : N bilangan asli

Output : A

<u>Langkah-langkah :</u>

1. Nilai awal A = 1, B = 1
2. Hitung A = A*B, dan B=B+1
3. Jika B < N kembali ke langkah-2
4. Tulis A

5. Selesai

1.3. Sejarah Pekembangan bahasa Visual BASIC

1.3.1. Perkembangan Bahasa BASIC

Bill Gates, pendiri Microsoft, memulai bisnis perangkat lunak dengan mengembangkan interpreter bahasa BASIC (Beginner's All-purpose Symbolic Instruction Code) untuk Altair 8800, untuk kemudian ia ubah agar dapat berjalan di atas IBM PC dengan sistem operasi DOS. Perkembangan berikutnya ialah diluncurkannya BASICA (basic-advanced) untuk DOS. Setelah BASICA, Microsoft meluncurkan Microsoft QuickBasic dan Microsoft Basic (dikenal juga sebagai Basic Compiler).

Visual Basic adalah pengembangan dari bahasa komputer BASIC. Bahasa BASIC diciptakan oleh Professor John Kemeny dan Thomas Eugene Kurtz dari Perguruan Tinggi Dartmouth pada pertengahan tahun 1960-an. Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa digunakan oleh para programer untuk menulis program-program komputer sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer. Sejak saat itu, banyak versi BASIC yang dikembangkan untuk digunakan pada berbagai platform komputer, seperti Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, Apple BASIC dan lain-lain.

Apple BASIC dikembangkan oleh Steve Wozniak, mantan karyawan Hewlett Packard dan teman dekat Steve Jobs (pendiri

Apple Inc.). Steve Jobs pernah bekerja dengan Wozniak sebelumnya (mereka membuat game arcade “Breakout” untuk Atari). Mereka mengumpulkan uang dan bersama-sama merakit PC, dan pada tanggal 1 April 1976 mereka secara resmi mendirikan perusahaan komputer Apple. Popularitas dan pemakaian BASIC yang luas dengan berbagai jenis komputer turut berperan dalam mengembangkan dan memperbaiki bahasa itu sendiri, dan akhirnya berujung pada lahirnya Visual Basic yang berbasis GUI (Graphic User Interface) bersamaan dengan Microsoft Windows. Pemrograman Visual Basic begitu mudah bagi pemula dan programer musiman karena ia menghemat waktu pemrograman dengan tersedianya komponen-komponen siap pakai. Hingga akhirnya Visual Basic juga telah berkembang menjadi beberapa versi, sampai yang terbaru, yaitu Visual Basic 2010. Bagaimanapun juga Visual Basic 6.0 tetap menjadi versi yang paling populer karena mudah dalam membuat programnya dan ia tidak menghabiskan banyak memori.

Sejarah BASIC di tangan Microsoft sebagai bahasa yang diinterpretasi (BASICA) dan juga bahasa yang dikompilasi (BASCOM) membuat Visual Basic diimplementasikan sebagai gabungan keduanya. Programmer yang menggunakan Visual Basic bisa memilih kode bahasa pemrograman yang dikompilasi atau kode yang harus bahasa pemrograman yang diinterpretasikan sebagai hasil porting dari kode VB. Sayangnya, meskipun sudah terkompilasi jadi bahasa mesin, DLL bernama MSVBVMxx.DLL tetap dibutuhkan.

Namun karakteristik bahasa terkompilasi tetap muncul (ia lebih cepat dari kalau kita pakai mode terinterpretasi).

1.3.2. Perkembangan Bahasa Visual Basic

Visual Basic 1.0 dikenalkan pada tahun 1991. Konsep pemrograman dengan metode drag-and-drop untuk membuat tampilan aplikasi Visual Basic ini diadaptasi dari prototype generator form yang dikembangkan oleh Alan Cooper dan perusahaannya, dengan nama Tripod. Microsoft kemudian mengontrak Cooper dan perusahaannya untuk mengembangkan Tripod menjadi sistem form yang dapat diprogram untuk Windows 3.0, di bawah kode nama Ruby. Tripod tidak memiliki bahasa pemrograman sama sekali. Ini menyebabkan Microsoft memutuskan untuk mengkombinasikan Ruby dengan bahasa pemrograman Basic untuk membuat Visual Basic.

<u>Perkembangan dari waktu ke waktu</u>

- **20 Mei 1991**: Microsoft merilis Visual Basic versi 1.0 untuk Windows di Windows World 1991, Atlanta, Amerika Serikat. Programmer dapat membuat antar muka pengguna dengan mudah. Pembuatan aplikasi bisa lebih cepat dibandingkan sebelumnya. Kode ditulis oleh programmer untuk memberikan aksi terhadap Event dari pengguna (bagaimana pengguna meres
- **1 September 1992**: Microsoft mengumumkan Microsoft Visual Basic for MS-DOS® dalam edisi Standard dan Professional.

Seperti Visual Basic untuk Windows, versi ini mengkombinasikan kemudahan pendisainan secara grafis dengan kekuatan dan keunggulan dalam banyak hal di pemrograman

- **2 November 1992**: Microsoft mengumumkan kemampuan dari Visual Basic versi 2.0 untuk Windows pada edisi Professional dan Standard. Versi ini menyertakan lebih dari 300 fitur baru dan peningkatan untuk pengembangan aplikasi yang lebih cepat, pengaksesan ke fitur-fitur tingkat lanjut di Windows, dan produktivitas pengembang yang lebih besar lagi. Di antaranya: MDI Forms, ODBC, dan variabel
- **14 Mei 1993**: Microsoft mengumumkan Visual Basic versi 3.0 untuk edisi Standard dan Professional. Versi 3.0 menyediakan kemudahan akses ke berbagai sumber data yang banyak dengan mengintegrasikan mesin database Microsoft Access Database for Windows 1.1 dan kemampuan di bidang aplikasi melalui Object Linking and Embedding (OLE) 2.0. Juga ditambahkan tools baru berupa controls baru, penggunaan yang lebih mudah, dan sebuah peningkatan standarisasi control bagi.
- **29 Juni 1993**: Microsoft mengumumkan bahwa Microsoft Visual Basic for Applications (VBA) akan diintegrasikan ke dalam Microsoft Excel 5.0 dan Microsoft Project 4.0. Visual Basic akan melayani bahasa Macro yang umum di versi yang akan datang dari aplikasi-aplikasi untuk Windows dan Macintosh. Fitur Macro yang umum untuk aplikasi dan OLE 2.0, keduanya merupakan elemen penting di visi dan strategi

Microsoft untuk aplikasi yang dapat diprogram —yang akan memberikan kemampuan bagi pengguna untuk mengintegrasikan aplikasi-aplikasi, tugas-tugas otomatis, dan pembuatan solusi yang bersifat kostumais. VBA memiliki mesin pengembangan tangguh dan fleksibel yang sama yang membuat Visual Basic menjadi terkenal, teknologi pemrograman yang memperoleh

- **14 November 1994**: Visual Basic versi 4.0 untuk Windows didemonstrasikan di Fall/COMDEX '94 di Las Vegas. Demonstrasi ini berfokus pada perannya sebagai aplikasi berbasis Windows 32-bit pertama yang menggunakan kontrol kostumais OLE (OCXs) — komponent perangkat lunak yang dapat digunakan kembali yang ditetapkan dengan spesifikasi OLE. Pada versi ini diperkenalkan: Class, OXC's, dan programmer dapat membuat add-ins sendiri dengan
- **12 September 1995**: Visual Basic versi 4.0 untuk MS-DOS, Microsoft Windows NT®, dan Windows 95 diumumkan. Untuk meningkatkan edisi Standard dan Professional, maka Edisi Enterprise yang baru diperkenalkan untuk memenuhi kebutuhan perusahaan dan team pengembang. Semua versi menyertakan dukungan teknologi OLE yang lebih dikembangkan, perbaikan akses data, sebuah lingkungan pengembangan yang bersifat visual, dan migrasi ke Windows 95 dan Windows NT untuk aplikasi-aplikasi yang sudah
- **7 Desember 1995**: Microsoft mengumumkan Microsoft Visual Basic Scripting Edition (VBScript), sebuah bahasa skrip Internet

berbasiskan pada Visual Basic. VBScript merupakan sebuah skrip dengan performansi yang tinggi yang didisain untuk membuat isi yang bersifat aktif pada suatu situs. VBScript membolehkan pengembang untuk mengaitkan dan mengotomatiskan berbagai jenis object di halaman situs, termasuk object OLE. VBScript menjadi sebuah spesifikasi yang diusulkan berlisensi gratis yang tersedia bagi komunitas

- **3 Februari 1997**: Microsoft membuka sistem pemrograman Visual Basic versi 5.0, Edisi Professional — tool versi terakhir yang paling terkenal di dunia untuk membangun aplikasi yang tangguh (Rapid Application Development RAD) tool. Fitur-fitur Visual Basic versi 5.0 terdiri dari sejumlah performansi yang signifikan dan peningkatan produktivitas yang akan membuat para pengembang lebih efisien dan fleksibel serta cepat dalam memberikan solusi bagi pengguna akhir. Fitur-fitur tingkat lanjut — seperti kompilasi berjenis native code, akses database berkecepatan tinggi, dan sebuah lingkungan pengembangan yang ditingkatkan — membuat Visual Basic versi 5.0 menjadi versi yang paling hebat. Sebagai tambahan, kemampuan teknologi pembuatan komponen Microsoft ActiveX® di Visual Basic versi 5.0 akan membolehkan lebih dari 3 juta pengembang menggunakan Visual Basic dalam membuat aplikasi yang berbasis pada komponen yang bertujuan untuk digunakan di Internet, intranets, dan lingkungan tradisional client/
- **10 Maret 1997**: Versi terakhir yang dirilis dari Visual Basic 5.0, Edisi Pembuatan Kontrol tersedia. Edisi Pembuatan Kontrol

adalah cara mudah untuk membuat ActiveX Controls bagi Internet, intranets, dan aplikasi client/

- **15 Juni 1998**: Microsoft mengumumkan Visual Basic versi 6.0, dan dimasukkan ke dalam Microsoft Visual Studio® versi 6.0. Fitur-fitur Visual Basic versi 6.0 menyediakan pengaksesan data secara terintegrasi dan bersifat grafis ke sumber data (data source) ODBC atau OLE DB manapun, dan perangkat tambahan database yang didisain untuk database Oracle dan Microsoft SQL Server™. Fitur unggulan di versi ini adalah: ActiveX Data Objects (ADO) untuk memanipulasi dan membuat database. Fitur Pengembangan Situs membawa kemudahan dalam penggunaan, model pemrograman berbasis komponen dari Visual Basic untuk membuat HTML – dan Dynamic HTML (DHTML) – berbasis aplikasi. Fitur-fitur baru ini — dikombinasikan dengan optimisasi performansi, pengembangan aplikasi yang disederhanakan dan debugging, dan dukungan untuk Microsoft teknologi server — membuat Visual Basic versi 6.0 sebuah pilihan yang ideal untuk membangun aplikasi berskala
- **2 September 1998**: Visual Studio 6.0, solusi tool pengembangan lengkap berskala perusahaan tersedia. Visual Studio 6.0 menyertakan Visual Basic versi 6.0, Microsoft Visual C++®, Microsoft Visual FoxPro®, Microsoft Visual InterDev® Web, dan Microsoft Visual J++®. Bersamaan dengan diluncurkannya Developer Days ’98, lebih dari 115

perusahaan papan atas mengumumkan dukungan terhadap Visual Studio 6.

- **28 April 1999**: Lingkungan pengembangan VBA versi 6.0 dan Software Development Kit (SDK) VBA (SDK) versi 6.0 tersedia bagi vendor perangkat lunak pihak ketiga melalui program lisensi VBA. VBA 6.0 merupakan sebuah teknologi pengembangan yang tangguh untuk aplikasi yang dikemas secara kostumais dan merupakan sebuah komponen utama di Microsoft Office
- **15 Februari 2000**: Steve Ballmer, presiden yang baru saja ditunjuk dan CEO dari Microsoft Corporation, berbicara kepada lebih dari 2.000 pengembang dalam pidatonya pada Visual Basic Insiders Technical Summit (VBITS), menyatakan visi Microsoft terhadap situs yang dapat diprogram dan memaksa kembali komitmen perusahaan yang dalam terhadap para pengembang Visual
- **11 Juli 2000**: Microsoft menyampaikan Tool Platform .NET untuk XML Web Services. Dalam pidatonya yang ditujukan pada acara Microsoft Professional Developers Conference (PDC) 2000 yang kedelapan, Paul Maritz, wakil presiden kelompok dari Platforms Group di Microsoft, memperlihatkan Microsoft .NET Framework dan Microsoft Visual Studio .NET. Ini merupakan versi terakhir dari perangkat pengembangan yang digunakan di seluruh dunia, Visual Studio .NET menyediakan dukungan untuk pengembangan drag-and-drop terhadap XML Web Services. Secara bersama, kedua produk tersebut

menyediakan produktivitas yang tinggi kepada para pengembang, suatu lingkungan yang terdiri dari banyak bahasa untuk pembangunan, pengantaran, dan pengintegrasian XML Web Services pada platform Microsoft .

- **13 November 2000**: Microsoft mengumumkan kemampuan Visual Studio .NET versi Beta 1. Dalam pidato acara COMDEX/Fall 2000, Bill Gates mengumumkan kemampuan versi beta pertama Visual Studio .NET dan .NET Framework, dua teknologi kunci untuk memungkinkan para pengembang membangun XML Web services pada platform .NET. XML Web services merupakan aplikasi dan komponen yang dibuat tersedia melewati situs dengan menggunakan XML dan Simple Object Access Protocol (SOAP), dan kunci yang dapat diprogram untuk membangun generasi masa depan Internet. Microsoft membuat Visual Studio .NET Beta 1 dan .NET Framework tersedia bagi jutaan pelanggan dan rekanan
- **13 Februari 2002**: Microsoft mengumumkan kemampuan dari Visual Studio.NET versi akhir.

Latihan

Misalkan kita ingin menentukan jenis dan akar-akar persamaan kuadrat, jika diketahui persamaan kuadratnya. Tentukan :

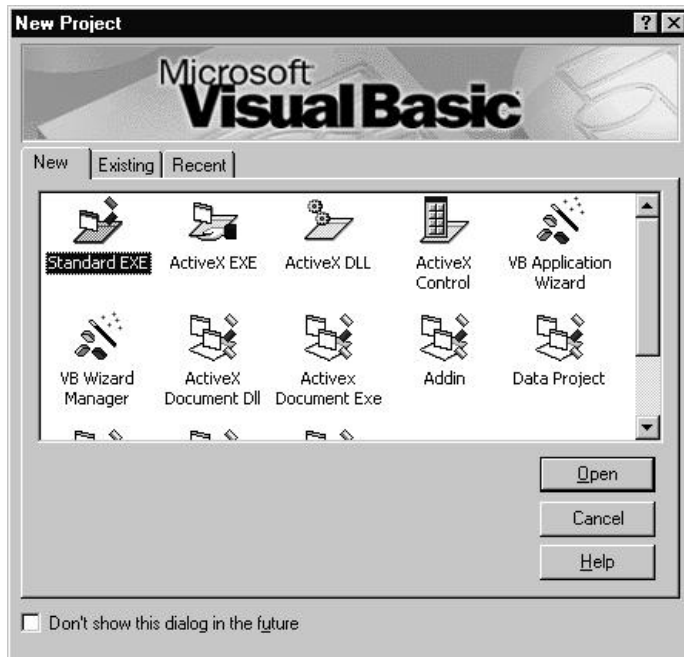
1. Diagram alurnya!
2. Algoritmanya!

Bab 2
Dasar Pemrograman Visual Basic

Visual Basic merupakan bahasa yang mendukung Pemrograman berorientasi objek. Kita sangat muda meletakkan komponen-komponen yang kita inginkan pada Form Designer. Visual Basic langsung menyediakan jendela/ window code yang dapat dilengkapi dengan mudah. Pertama kita akan mempelajari bagaimana cara Visual Basic dan mengamati apa yang ada di jendela Form Designer. Selanjutnya perbedaan penulisan operasi, relasi, dan logika antara matematika dan Visual Basic, serta bagaimana penulisan fungsi sederhana. Akhirnya pembuatan proyek pemrograman sederhana akan dipelajari.

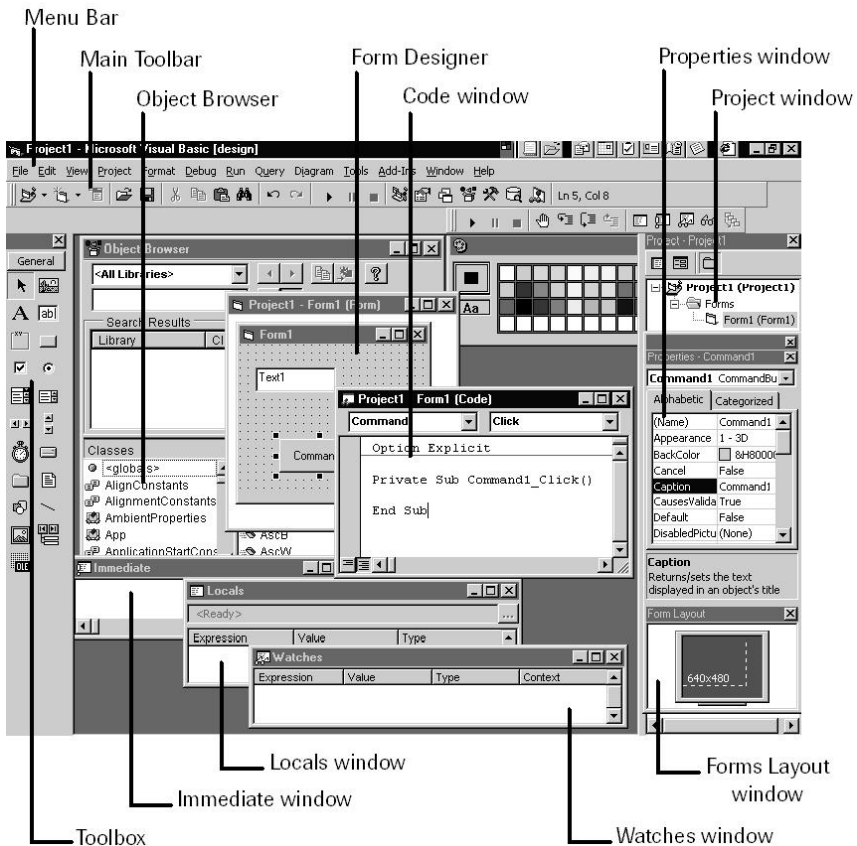
2.1. Masuk Visual Basic

Salah satu cara masuk ke Visual Basic adalah menjalankannya dari **Menu Start**, pilih **All Programs**, dan pilih **Microsoft Visual Basic 6.0**. Selanjutnya anda akan melihat jendela **New Project**

seperti Gambar 2.1. Untuk awal pembelajaran pada Mata Kuliah ini, pilih project **Standard EXE**

Gambar 2.1. New Project

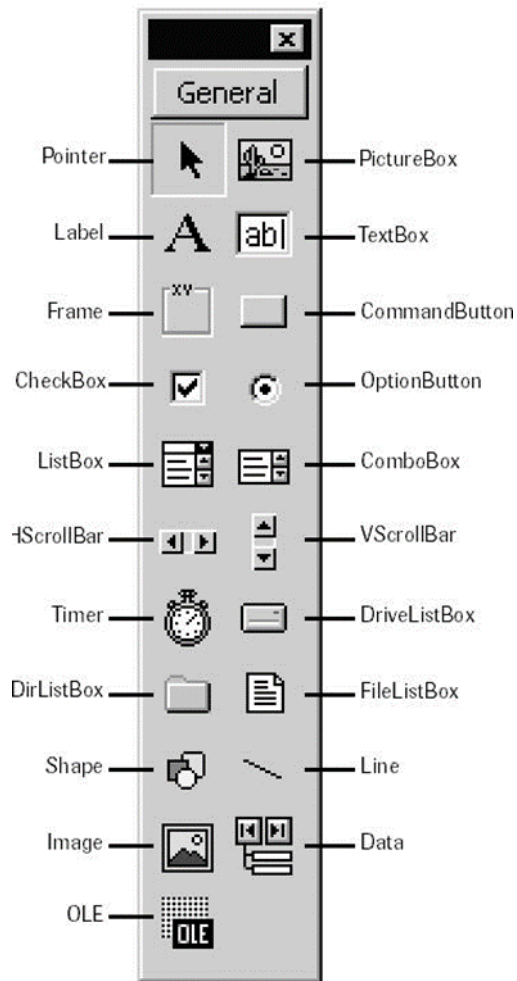
Jendela Visual Basic 6 menggunakan model Multiple Document Interface (MDI). Bagian dan nama-nama objek yang ada pada jendela tersebut dapat dilihat pada Gambar 2.2. Nama-nama objek yang ada pada jendela tersebut dapat diketahui dengan cara meletakkan pointer mouse di atas objek tersebut maka nama objek akan muncul.

Dari begitu banyak objek yang tersedia pada jendela tersebut, kita akan focus untuk beberapa hal berikut:

1. **Menu Bar**, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll.

Gambar 2.2. Jendela Visual Basic

2. **Main Toolbar**, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
3. **Jendela Project**, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan Ctrl+R untuk menampilkan jendela project, ataupun menggunakan icon Project Explorer.
4. **Jendela Form Designer**, jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini menyerupai kanvas bagi seorang pelukis.
5. **Jendela Toolbox**, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
6. **Jendela Code**, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi Shift-F7.
7. **Jendela Properties**, merupakan daftar properti-properti object yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela properti.
8. **Jendela Color Palette**, adalah fasilitas cepat untuk mengubah warna suatu object.
9. **Jendela Form Layout**, akan menunjukan bagaimana form bersangkutan ditampilkan ketika runtime.

2.2. Toolbox

Jendela Toolbox merupakan jendela yang sangat penting untuk melengkapi Form designer seperti yang kita inginkan. Dari jendela ini anda dapat mengambil komponen-komponen (object) yang akan ditanamkan pada form untuk membentuk user interface. Beberapa komponen pada jendela Toolbox dapat dilihat pada Gambar 2.3. Jika anda membutuhkan komponen lain yang mungkin diperlukan maka anda dapat mengambilnya dengan cara klik **Project** pada menu Bar dan selanjutnya klik **Componets**.

Gambar 2.3 Toolbox

Penjelasan singkat untuk komponen-komponen tersebut adalah sebagai berikut:

1. ***Pointer*** bukan merupakan suatu kontrol; gunakan icon ini ketika anda ingin memilih kontrol yang sudah berada pada form.
2. ***PictureBox*** adalah kontrol yang digunakan untuk menampilkan image/ grafik dengan format: BMP, DIB (bitmap), ICO (icon), CUR (cursor), WMF (metafile), EMF (enhanced metafile), GIF, dan JPEG.
3. ***Label*** adalah kontrol yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.
4. ***TextBox*** adalah kontrol yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal, atau banyak baris.
5. ***Frame*** adalah kontrol yang digunakan sebagai kontainer bagi kontrol lainnya.
6. ***CommandButton*** merupakan kontrol hampir ditemukan pada setiap form, dan digunakan untuk membangkitkan event proses tertentu ketika pemakai melakukan klik padanya.
7. ***CheckBox*** digunakan untuk pilihan yang isinya bernilai yes/no, true/false.
8. ***OptionButton*** sering digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
9. ***ListBox*** mengandung sejumlah item, dan user dapat memilih lebih dari satu (bergantung pada property

MultiSelect).

10. ***ComboBox*** merupakan konbinasi dari TextBox dan suatu ListBox dimana pemasukkan data dapat dilakukan dengan pengetikkan maupun pemilihan.
11. ***HScrollBar*** dan ***VScrollBar*** digunakan untuk membentuk scrollbar berdiri sendiri.
12. ***Timer*** digunakan untuk proses background yang diaktifkan berdasarkan interval waktu tertentu. Merupakan kontrol non-visual.
13. ***DriveListBox***, ***DirListBox***, dan ***FileListBox*** sering digunakan untuk membentuk dialog box yang berkaitan dengan file.
14. ***Shape*** dan ***Line*** digunakan untuk menampilkan bentuk seperti garis, persegi, bulatan, oval.
15. ***Image*** berfungsi menyerupai image box, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa kontrol image menggunakan resource yang lebih kecil dibandingkan dengan PictureBox
16. ***Data*** digunakan untuk *data binding*
17. ***OLE*** dapat digunakan sebagai tempat bagi program eksternal seperti Microsoft Excel, Word, dll.

2.3. Operasi Matematika dan Fungsi

Dalam pembuatan code program, kita akan menggunakan operasi, relasi, dan nilai logika. Perbandingan penulisan simbol pada matematika dengan Visual Basic mempunyai sedikit perbedaan.

Aritmetika

No	Uraian	Matematika	Visual Basic
1	Negatif	-a	-a
2	Jumlah	a + b	a + b
3	Kurang	a – b	a – b
4	Kali	a × b	a * b
5	Bagi	a : b	a / b
6	Pangkat	a^b	a^b

Relasi

No	Uraian	Matematika	Visual Basic
1	Sama	a = b	a = b
2	Tidak sama	a ≠ b	a <> b
3	Kurang dari	a < b	a < b
4	Lebih dari	a > b	a > b
5	Kurang atau sama	a < b	a <= b
6	Lebih atau sama	a > b	a >= b

Logika

No	Uraian	Matematika	Visual Basic
1	Tidak	~a	Not a
2	Dan	a ∧ b	a And b

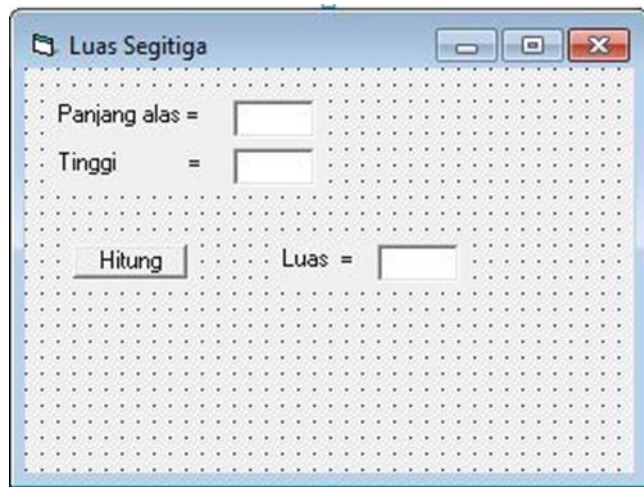
3	Atau	$a \vee b$	a Or b
4	Atau Ekslusiv	a XOr b	a XOr b

Selain mengenal operasi matematika, kita juga perlu mengenal beberapa fungsi yang sering digunakan dalam pemrograman, antara lain:

1. sin(bilangan) : menghitung nilai fungsi sinus.
2. cos(bilangan) : menghitung nilai fungsi cosinus.
3. tan(bilangan) : menghitung nilai fungsi tangen.
4. round(bilangan,[berapa dibelakang koma]) : membulatkan bilangan berapa di belakang koma.
5. log(bilangan) : menghitung nilai logaritma natural.
6. abs(bilangan) : menghitung nilai mutlak.
7. str(bilangan) : mengubah bilangan menjadi string.
8. val(string): mengubah string menjadi bilangan real.

2.4. Proyek Sederhana

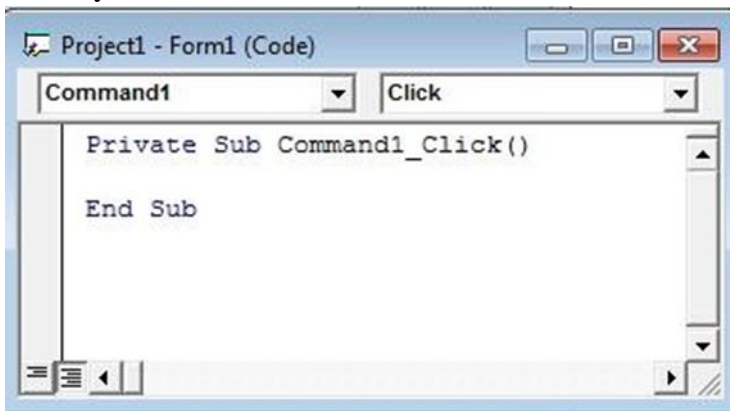
Pada kesempatan ini kita akan membuat proyek menghitung luas segitiga jika diketahui panjang alas dan tingginya. Kita akan membuat bentuknya seperti Gambar 2.4. Sebelumnya, buat folder Bab_02_01 sebagai tempat penyimpanan proyek ini!

kita akan menggunakan operasi, relasi, dan nilai logika. Perbandingan penulisan simbol pada matematika dengan Visual Basic mempunyai sedikit perbedaan.

Gambar 2.4 Form Designer dan Objek-Objek

No	Tools	Properti	Nilai
1	Form	Name Caption	Form1 Luas Segitiga
2	Label	Name Caption	Label1 Panjang alas =
3	TextBox	Name Text	Text1
4	Label	Name Caption	Label2 Tinggi =
5	TextBox	Name Text	Text2
6	Label	Name Caption	Label3 Luas =

7	TextBox	Name Text	Text3
8	CommandButton	Name Caption	Command1 Hitung

Caranya:

1. Pilih komponen **Label** pada ToolBox, setelah itu klik seret mouse pada Form Designer pada tempat yang diinginkan. Selanjutnya perhatikan Propertinya, terutama nilai dari **Name** dan **Caption**, Ganti nilai Caption = Panjang alas =, sedangkan nilai untuk **Name** tidak perlu diubah.
2. Pilih komponen TextBox pada ToolBox, setelah itu klik seret mouse pada Form Designer pada tempat yang diinginkan. Selanjutnya perhatikan Propertinya, terutama nilai dari **Name** dan **Text**, Ganti nilai Text = (kosong), sedangkan nilai untuk **Name** tidak perlu diubah.
3. Dengan cara yang sama lanjutkan untuk Label dan TextBox berikutnya.

Gambar 2.5 Jendela Code

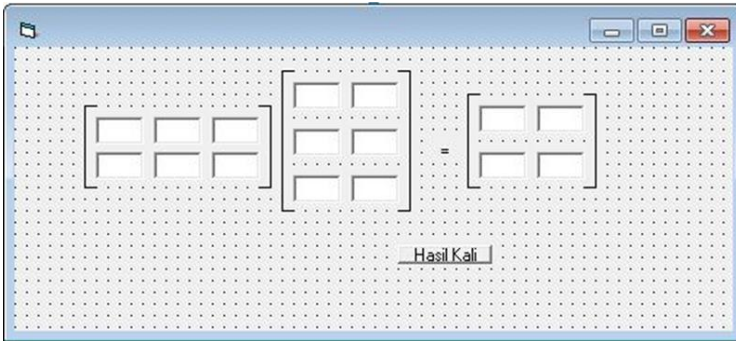
4. Pilih komponen **CommandButton** pada ToolBox, setelah itu klik seret mouse pada Form Designer pada tempat yang diinginkan. Selanjutnya perhatikan Propertinya, terutama nilai dari **Name** dan **Caption**, Ganti nilai Caption = **Hitung**, sedangkan nilai untuk **Name** tidak perlu diubah. Setelah itu double klik pada tulisan Hitung untuk masuk pada jendela Code sehingga akan tampak Gambar 2.5. Dengan tindakan Click maka semua perintah diantara **Private ...... End** akan dikerjakan. Karena kita ingin bahwa jika tombol Hitung diklik maka nilai Text untuk Text3 berubah menjadi hasil luas segitiga maka perintah yang ditulis adalah:

```
Private Sub Command1_Click()
Text3.Text = Text1.Text * Text2.Text / 2
End Sub
```

5. Save proyek ini pada folder Bab_02_01.
6. Untuk menjalankan proyek yang baru selesai, pilih Run di Menu Bar dan klik Start, atau boleh langsung tekan F5 pada keyboard. Ketik nilai untuk panjang alas, tinggi, dan selanjutnya klik tombol Hitung! Amati apa yang terjadi!

Seandainya anda ingin menggunakan hasil pekerjaan ini untuk keperluan lain yang berdiri sendiri maka anda dapat melakukannya dengan memilih **File** pada Menu Bar, dan klik **Make ......exe**. Selanjutnya lengkapi apa yang diminta, dan akhirnya file yang tercipta dapat digunakan tanpa membutuhkan software Visual Basic, Selamat mencoba!!!

Latihan

Buatlah proyek hasil kali dua matriks (misalnya A×B = C) yang bentuknya seperti Gambar 2.5. Save proyek ini di folder Bab_02_02.

Gambar 2.5 Form Designer dan Objek-Objek Perkalian Matriks

Bab 3
Kontrol Program If

Dalam Bab ini kita akan berbicara tentang bagaimana menggunakan bahasa pemrograman tentang kontrol program if. Kadang kita hanya ingin menugaskan komputer melakukan perintah hanya jika suatu kondisi tertentu dipenuhi, sedangkan dalam kasus lain diabaikan. Hal seperti ini akan kita bahas dalam Sub Bab 3.1. Sedangkan untuk dua Sub Bab berikutnya kita akan membahas lebih dari satu percabangan. Setelah selesai mempelajari Bab ini mahasiswa diharapkan mampu menggunakan dan memilih kontrol program if dalam penulisan kode Visual Basic yang sesuai dengan keperluan suatu proyek.

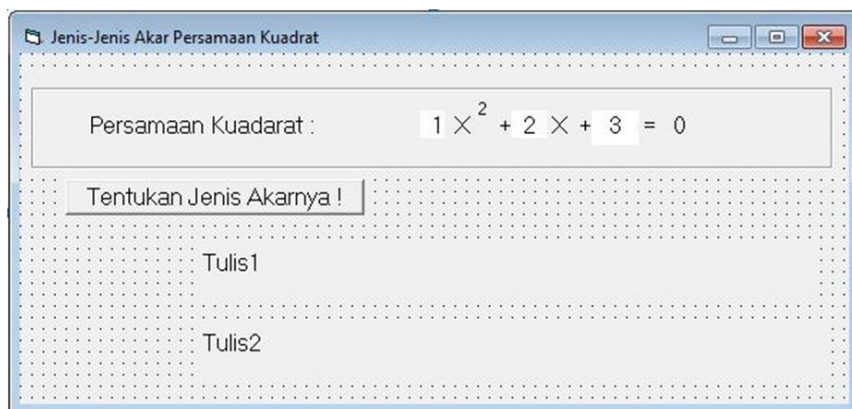
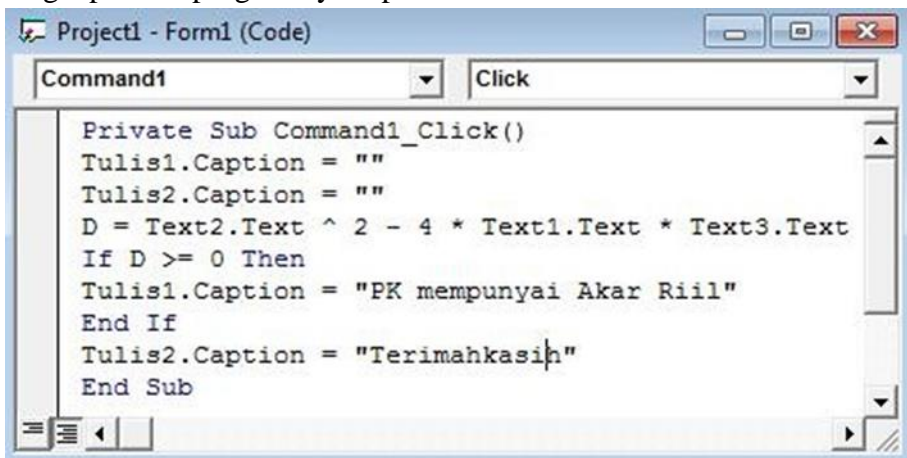
3.1. Sintaks *If ... Then*

Kontrol program *If* (jika) akan mengatur komputer seperti arti dari kata tersebut, yaitu jika ekspresi yang mengikuti kontrol program *If* benar perintah setelah *Then* sampai batas *End* akan dilaksankan. Jika ekspresi salah maka komputer tidak akan

mengerjakan perintah-perintah antara *Then* dan *End* tetapi langsung melanjutkan perintah setelah *End*. Sistimatika sintaksnya adalah :

```
If ekspresi Then
Perintah-perintah
End If
```

Contoh 1

Kita akan membuat proyek dengan Form Desiger seperti Gambar 3.1 berikut:

Gambar 3.1 Form Jenis-Jenis Akar Persamaan Kuadrat

Dengan ketentuan sebagai berikut:

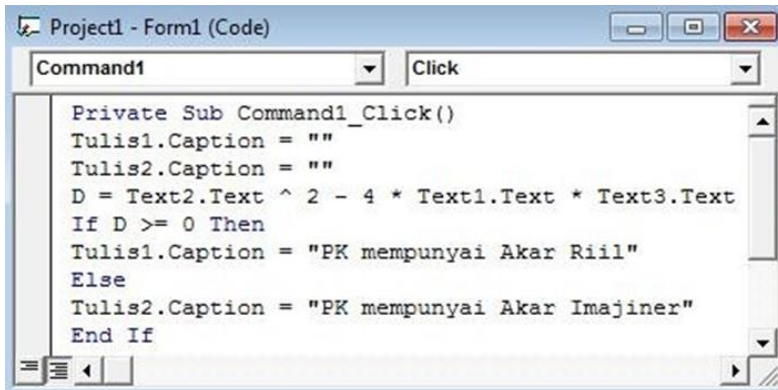
No	Tools	Properti	Nilai
1	TextBox	Name Text	Text1 1
2	TextBox	Name Text	Text2 2

3	TextBox	Name Text	Text3 3
4	CommandButton	Name Caption	Command1 Tentukan Jenis Akarnya!
5	Label	Name Caption	Tulis1 Tulis1
6	Label	Name Caption	Tulis2 Tulis2

Nama tools yang lain, terserah anda. Selanjunya Double Klik *CommandButton Command1* (Tentukan Jenis Akarnya!) dan lengkapi kode programnya seperti berikut:

Simpan Proyek ini pada folder **Bab_03_01** dengan nama: **Jenis Akar** (caranya: Klik **File → Save Project As...**, pilih di folder mana akan disimpan, tuliskan nama project-nya: **Jenis Akar 31**, dan

akhirnya klik **Save**). Perhatikan perintah yang tertulis di antara sintaks **then** sampai dengan **End If**. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai a, b, dan c yang berbeda sehingga D > 0, atau D < 0. Amati apa yang terjadi! Akhirnya berikan kesimpulan!

3.2. Sintaks *If ... Then ... Else*

Dengan menggunakan sintaks ini, kita diberi kesempatan memerintah komputer untuk kondisi ekspresi benar dengan suatu perintah, sedangkan untuk kondisi salah dengan perintah yang lain. Sistimatika sintaksnya adalah :

```
If ekspresi Then
Perintah-perintah 1
Else
Perintah-perintah 2
End If
```

Contoh 2

Proyek yang akan kita gunakan adalah proyek yang ada di sub bab sebelumnya, tetapi program akan memberitahu apakah akarnya persamaan kuadratnya riil atau imajiner. Anda boleh meng-copy folder sebelumnya, mem-paste dan selanjutnya mengganti nama foldernya menjadi **Bab_03_02**. Buka buka folder baru ini dan klik file yang ada sebelumnya, yaitu nama file: **Jenis Akar 31**. Untuk membuat proyek lain, kita akan menyimpan proyek ini dengan nama: Jenis Akar 32. Caranya: **Klik File → Save Project As...**, pilih folder **Bab_03_02** tempat file akan disimpan, tuliskan nama project-nya: **Jenis Akar 32**, dan akhirnya klik **Save**. Selanjutnya Selanjunya

Double Klik *CommandButton Command1* (Tentukan Jenis Akarnya!) dan lengkapi kode programnya seperti berikut:

Perhatikan perintah yang tertulis di antara sintaks **then** sampai dengan **Else**. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Perhatikan perintah yang tertulis di antara sintaks **Else** sampai dengan **End If**. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Selanjutnya jalankan programnya untuk berbagai nilai a, b, dan c yang berbeda sehingga D > 0, atau D < 0. Amati apa yang terjadi! Akhirnya berikan kesimpulan!

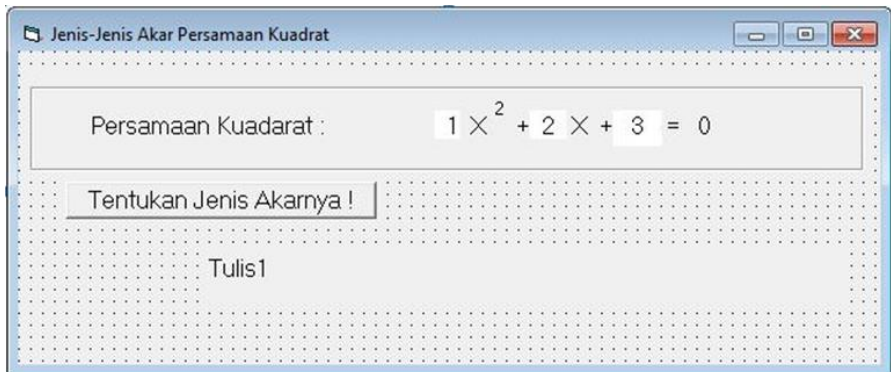
3.3. Sintaks If ... Then ... ElseIf ... Else

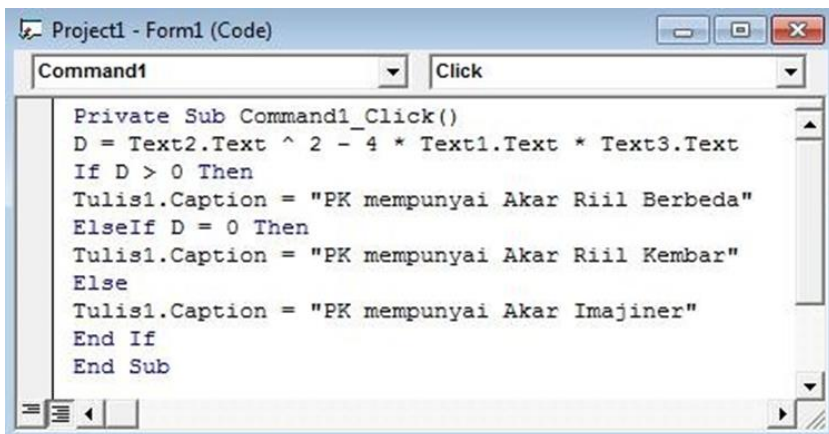
Jika kita membutuhkan percabangan pekerjaan lebih dari dua bagian maka kita dapat menggunakan sintaks **If ... Then ... ElseIf ...**

Else. Bahkan ElseIf dapat digunakan berulang-ulang sesuai kebutuhan. Sistimatika sintaksnya adalah seperti berikut:

```
If ekspresi1 Then
Perintah-perintah 1
ElseIf ekspresi2 Then
Perintah-perintah 2
...
[ElseIf ekspresi-n Then
Perintah-perintah n]
Else
Perintah-perintah (n+1)
End If
```

Contoh 3

Dengan cara yang sama buat folder penyimpanan proyek ini dengan nama **Bab_03_03**. Proyek yang akan kita gunakan adalah proyek yang ada di sub bab 3.1 sebelumnya, tetapi program akan memberitahu apakah akarnya persamaan kuadratnya bernilai riil berbeda, riil kembar, atau imajiner. Dalam proyek ini, jenis akar akan diberitahu hanya dalam satu label aja, sehingga label terakhir akan dihapus seperti Gambar 3.2.

Gambar 3.2 Jenis Akar Menggunakan ElseIf

Simpan proyek ini dengan nama: **Jenis Akar 33**. Selanjutnya Selanjunya Double Klik *CommandButton Command1* (Tentukan Jenis Akarnya!) dan lengkapi kode programnya seperti berikut:

Perhatikan perintah yang tertulis di antara sintaks **If...then** sampai dengan ElseIf. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Perhatikan perintah yang tertulis di antara sintaks **ElseIf...then** sampai dengan **Else**. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Perhatikan perintah yang tertulis di antara sintaks **Else** sampai dengan **End If**. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Selanjutnya jalankan programnya untuk berbagai nilai a, b, dan c yang berbeda sehingga $D > 0$, $D = 0$, atau $D < 0$. Amati apa yang terjadi! Akhirnya berikan kesimpulan!

Latihan

Buatlah proyek seperti proyek pada sub bab 3.3, tetapi program juga akan memberitahu nilai akar-akarnya kalau ada. Save proyek ini pada folder **Bab_03_04**.

Bab 4
Kontrol Program Select Case

Dalam Bab ini kita akan berbicara tentang bagaimana menggunakan bahasa pemrograman tentang kontrol program ***Case***. Pertama-tama kita akan memulai mempelajari sitimatika ***Select Case***, dan akhirnya kita akan menggunakan Tool ComboBox untuk contoh penggunaan *Select Case*. Setelah selesai mempelajari Bab ini mahasiswa diharapkan mampu menggunakan kontrol program *Select Case* dalam penulisan kode Visual Basic.

4.1. Sintaks Select Case

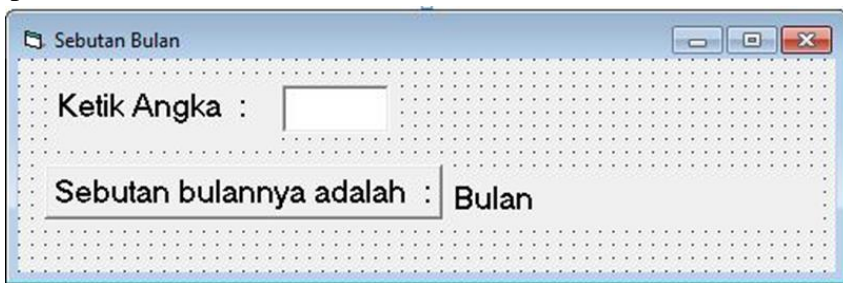
Sebenarnya kontrol program ***Select Case*** adalah penyederhanaan kontrol program ***if ... then ... elseif*** dengan ***elseif*** digunakan berulang-ulang. Sistimatika sintaksnya adalah :

```
Select Case Variabel
Case nilai_1:
Perintah-perintah_1
Case nilai_2:
Perintah-perintah_2
...
```

```
Case Else
Perintah-perintah_n
End Select
```

Jika nilai variabel sama dengan nilai sebelah kanan *Case* maka komputer mengerjakan perintah-perintah di bawah *Case* tersebut sampai *Case* berikutnya.

Contoh 1

Save proyek ini pada folder **Bab_04_01**. Kita akan membuat proyek penggunaan Select Case sederhana dengan Form Desiger seperti Gambar 4.1 berikut:

Gambar 4.1 Form Sebutan Bulan

Dengan ketentuan sebagai berikut:

No	Tools	Properti	Nilai
1	Label	Name Caption	Label1 Ketik Angka :
2	TextBox	Name Text	Angka
3	CommandButton	Name Caption	Command1 Sebutan bulannya adalah :

4	Label	Name Caption	Bulan Bulan

Selanjunya Double Klik CommandButton Command1 (Sebutan bulannya adalah :) dan lengkapi kode programnya seperti berikut:
Private Sub Command1_Click() x = Val(Angka.Text)

```
Select Case x Case 1:
Bulan.Caption = "Januari" Case 2:
Bulan.Caption = "Pebruari" Case 3:
Bulan.Caption = "Maret" Case 4:
Bulan.Caption = "April" Case 5:
Bulan.Caption = "Mei" Case 6:
Bulan.Caption = "Juni" Case 7:
Bulan.Caption = "Juli" Case 8:
Bulan.Caption = "Agustus" Case 9:
Bulan.Caption = "September" Case 10:
Bulan.Caption = "Oktober" Case 11:
Bulan.Caption = "Nopember" Case 12:
Bulan.Caption = "Desember" Case Else
Bulan.Caption = "Tidak ada bulan sesuai angka
tersebut" End Select
End Sub
```

Simpan Proyek ini dengan nama: **Sebutan Bulan**. Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai, Amati apa yang terjadi! Akhirnya berikan kesimpulan!

4.2. ComboBox dan Select Case

Buat folder untuk tempat save projek ini dengan nam folder **Bab_04_02**. Kita akan memperkenalkan Tool ComboBox yang sering kita jumpai dalam penggunaan komputer. Buatlah Form Designer seperti Gambar 4.2.

Gambar 4.2. Form Designer Geometri

Dengan ketentuan sebagai berikut:

1	Label	Name Caption	Label1 Pilih Yang Anda Inginkan!

2	ComboBox	Name Text List	Combo1 Combo1 Persegi Panjang ***(Tekan Ctrl Enter)*** Segitiga ***(Tekan Ctrl Enter)*** Tabung ***(Tekan Ctrl Enter)*** Kerucut
3	Label	Name Caption	Nama1 Panjang =
4	TextBox	Name Text	Text1
5	Label	Name Caption	Nama2 Lebar =
6	TextBox	Name Text	Text2
7	Command Button	Name Caption	Command1 &Hitung
8	Label	Name Caption	Nama3 Luas =
9	Label	Name Caption	Hasil Hasil

Simpan Proyek ini dengan nama: Geometri. Selanjunya Double Klik *ComboBox Combo1* dan lengkapi kode programnya seperti berikut:

```
Private Sub Combo1_Click()
x = Combo1.Text
Select Case x
Case "Persegi Panjang":
Nama1.Caption = "Panjang ="
```

```
Nama2.Caption = "Lebar         ="
Nama3.Caption = "Luas    =" Hasil.Caption = "..."
Case "Segitiga":
Nama1.Caption = "Alas    ="
Nama2.Caption = "Tinggi        ="
Nama3.Caption = "Luas    =" Hasil.Caption = "..."
Case "Tabung":
Nama1.Caption = "Jari-jari Alas ="
Nama2.Caption = "Tinggi  ="
Nama3.Caption = "Volume =" Hasil.Caption = "..."
Case "Kerucut":
Nama1.Caption = "Jari-jari Alas ="
Nama2.Caption = "Tinggi  ="
Nama3.Caption = "Volume =" Hasil.Caption = "..."
End Select
End Sub
```

Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Selanjunya Double Klik *CommandButton Command1* (Hitung) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
x = Combo1.Text
Select Case x
Case "Persegi Panjang":
LV = Text1.Text *
Text2.Text Hasil.Caption = "..."
Case "Segitiga":
LV = Text1.Text * Text2.Text / 2
Case "Tabung":
LV = 3.14 * Text1.Text ^ 2 * Text2.Text
Case "Kerucut"
```

```
LV = 3.14 * Text1.Text ^ 2 * Text2.Text / 3
End Select
Hasil.Caption = LV
End Sub
```

Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan program dan lengkapi data yang dibutuhkan, amati apa yang terjadi!

Latihan

Perbaiki proyek pada Gambar 4.2 sehingga lebih indah dilihat dan lebih mudah digunakan! Save proyek ini pada folder **Bab_04_03**.

Bab 5
Kontrol Program Loop
For ... Next

Dalam Bab ini kita akan berbicara tentang bagaimana menggunakan bahasa pemrograman Loop tentang kontrol program *For ... Next*. Pertama-tama kita akan memulai mempelajari sitimatika *For ... Next*, dan akhirnya kita akan menggunakan *Tool PictureBox* untuk contoh penggunaan *For ... Next*. Setelah selesai mempelajari Bab ini mahasiswa diharapkan mampu menggunakan kontrol program ***For ... Next*** dalam penulisan kode Visual Basic.

5.1. Sintaks *For ... Next*

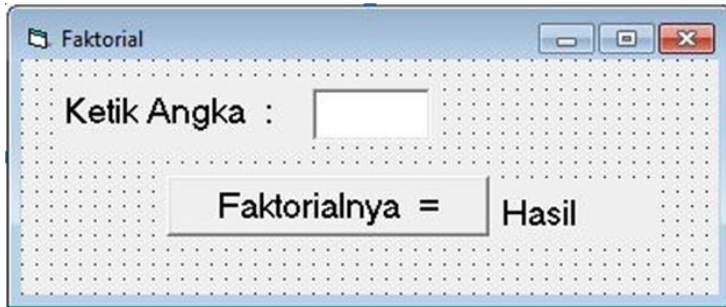
Loop adalah suatu kode program yang digunakan jika suatu pekerjaan akan dilakukan secara berulang-ulang. Salah satu sintaks loop adalah *Foor ... Next*. Sistimatika sintaksnya adalah :

```
For x = nilai_awal To nilai_akhir [Step langkah]
Perintah-perintah [memuat x]
Next [x]
```

Perintah-perintah yang ada diantara *For* sampai *Next* akan dilakukan berulang-ulang dengan mengganti nilai variabel *x* mulai

dari nilai_awal ditambah-tambah sebesar langkah jika masih nilai *x < nilai_akhir*.

Contoh 1

Buat folder penyimpanan proyek ini dengan nama **Bab_05_01**. Kita akan membuat proyek penggunaan *For ... Next* sederhana dengan Form Desiger seperti Gambar berikut:

Gambar 5.1 Form Faktorial

Dengan ketentuan sebagai berikut:

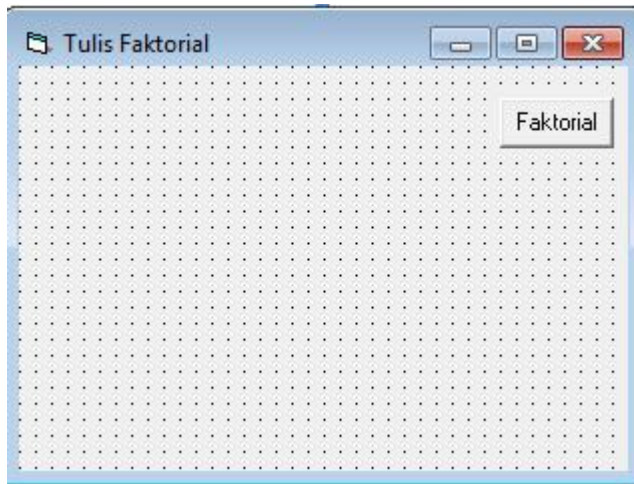
No	Tools	Properti	Nilai
1	Label	Name Caption	Label1 Ketik Angka :
2	TextBox	Name Text	Angka
3	CommandButton	Name Caption	Command1 Faktorialnya =
4	Label	Name Caption	Hasil Hasil

Selanjunya Double Klik *CommandButton Command1* (Faktorialnya =) dan lengkapi kode programnya seperti berikut:
Private Sub Command1_Click()

```
N = Angka.Text
A = 1
For i = 1 To N
A = A * i
Next Hasil.Caption = A
End Sub
```

Simpan Proyek ini dengan nama: **bab051**. Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai, amati apa yang terjadi! Akhirnya berikan kesimpulan!

Contoh 2

Buat folder penyimpanan proyek ini dengan nama **Bab_05_02.** Kita akan membuat proyek penggunaan *For ... Next* sederhana dengan Form Desiger seperti Gambar dengan cara menulis hasilnya langsung pada Form berikut:

Gambar 5.2 Form Tulis Faktorial

Dengan ketentuan sebagai berikut:

No	Tools	Properti	Nilai
1	CommandButton	Name Caption	Command1 Faktorial

Selanjunya Double Klik *CommandButton* *Command1* (Faktorial) dan lengkapi kode programnya seperti berikut:

Private Sub Command1_Click()

```
N = InputBox("Ketik bilangan asli = ")
A = 1
For i = N To 1 Step -1
A = A * i
Print A Next
Print "Jadi :   " & N & "! = " & A
End Sub
```

Simpan Proyek ini dengan nama: **bab052**. Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai, amati apa yang terjadi! Akhirnya berikan kesimpulan!

5.2. FictureBox dan *For ... Next*

Buat folder penyimpanan proyek ini dengan nama **Bab_05_03**. Kita akan memperkenalkan Tool FictureBox sebagai tempat gambar dan grafik. Buatlah Form Designer seperti Gambar 5.3! Dengan ketentuan sebagai berikut:

No	Tools	Properti	Nilai
1	CommandButton	Name Caption	Command1 Lukis
2	PictureBox	Name	Picture1

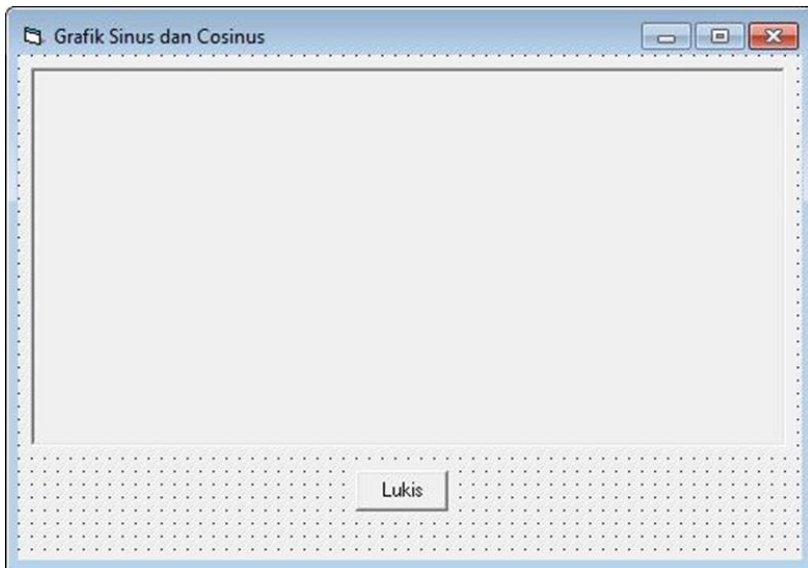
Simpan Proyek ini dengan nama: bab053. Selanjunya Double Klik *Command1* (Lukis) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
Picture1.ScaleMode = 0
Picture1.ScaleWidth = 360
Picture1.ScaleHeight = 2
Picture1.ScaleLeft = 0
Picture1.ScaleTop = -1
Picture1.Line (0, 0)-(360, 0)
For i = 0 To 360
Picture1.PSet (i, -Sin(i*3.14/180))
Picture1.PSet (i, -Cos(i*3.14/180))
Next i
```

```
End Sub
```

Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan!

Selanjutnya jalankan program, amati apa yang terjadi!

Anda juga dapat menggambar grafik berwarna dengan mengatur properties-nya, yaitu: ForeColor

Gambar 5.3. Form Grafik

Latihan

Buat proyek menggambar grafik parabola $y = ax^2 + bx + c$, dimana a, b, dan c diisi oleh pengguna. Usahakan titik puncak parabola dan titik potong terhadap sumbu-y terlihat! Buat folder penyimpanan proyek ini dengan nama **Bab_05_04**.

Bab 6
Kontrol Program Loop Do While

Dalam Bab ini kita akan berbicara tentang bagaimana menggunakan bahasa pemrograman dengan kontrol program *Do While ... Loop*. Kontrol program ini hampir sama dengan kode program *For ... Next* yang digunakan untuk pekerjaan berulang-ulang. Jika kita tahu berapa kali berulang maka kontrol program *For ... Next* lebih baik digunakan. Tetapi banyak pekerjaan tidak diketahui berapa kali diulang dan hanya berhenti jika kondisi tertentu dipenuhi. Untuk kasus seperti ini kita harus menggunakan kontrol program *Do While ... Loop*. Sistimatika sintaksnya ada dua, yaitu: *Do While ... Loop*, dan *Do ... Loop While*. Pertama-tama kita akan memulai mempelajari sitimatika *Do While ... Loop*, dan akhirnya kita akan menggunakan sistimatika kedua. Setelah selesai mempelajari Bab ini mahasiswa diharapkan mampu menggunakan dan membedakan dua sistimatika kontrol program *Do While ... Loop* dalam penulisan kode Visual Basic

6.1. Sintaks *Do While ... Loop*

Loop yang ini mengerjakan perintah antara *While* sampai dengan *Loop* berulang- ulang selama syarat yang yang ditetapkan masih benar. Sistimatika sintaks pertama adalah :

```
Do While kondisi
Perintah-Perintah
Loop
```

Jadi pertama-tama komputer akan memeriksa nilai dari *kondisi*, jika benar maka komputer akan mengerjakan perintah-perintah yang ada. Setelah menemukan *Loop* maka komputer akan kembali ke atas dan memeriksa nilai dari *kondisi*. Jika masih benar maka pekerjaan yang sama akan dikerjakan, tetapi jika sudah bernilai salah maka komputer akan melanjutkan pekerjaan di bawah *Loop*.

Contoh 1

Buat folder penyimpanan proyek ini dengan nama **Bab_06_01**. Kita akan membuat proyek penggunaan *Do While ... Loop* sederhana dengan Form Designer seperti Gambar 5.1 pada Bab 5 dan hanya berbeda pada kode programnya.

Selanjunya Double Klik *CommandButton Command1* (Faktorialnya =) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
N = Angka.Text
Dim A As Double
Dim i As Integer
A = 1
i = 1
```

```
Do While i <= N
A = A * i
i = i + 1
Loop
Hasil.Caption = A
End Sub
```

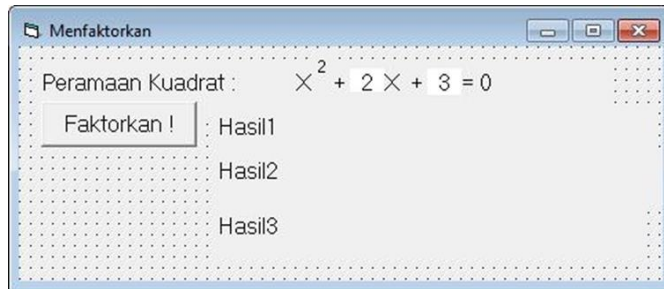
Simpan Proyek ini dengan nama: **Bab061**. Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai, amati apa yang terjadi! Akhirnya berikan kesimpulan!

Sistimatika sintaks kedua adalah:

```
Do
Perintah-Perintah
Loop While kondisi
```

Untuk sintaks seperti ini, komputer langsung mengerjakan perintah diantara *Do* dan *Loop*. Setelah sampai pada *Loop*, komputer memeriksa kondisi yang ada apakah benar atau salah. Jika benar akan kembali ke *Do*, tetapi jika salah maka komputer langsung mengerjakan perintah di bawah sintaks *Loop*. Berarti dengan sisntaks terakhir ini, komputer akan mengerjakan perintah minimal satu kali.

6.2. Proyek dengan Sintaks Do While ... Loop

Buat folder penyimpanan proyek ini dengan nama **Bab_06_02**. Kita akan membuat proyek pemaktoran persamaan kuadrat yang koefisien X^2 selalu satu dengan penggunaan *Do ... Loop While* yang Form Designer seperti Gambar 6.1 berikut:

Gambar 6.1 Form Designer Menfaktorkan

Seperti biasa, gunakan beberapa Label untuk membuat persamaan kuadrat dengan dua *TextBox*. Kedua *TextBox* dan Tool lain dengan ketentuan sebagai berikut:

No	Tools	Properti	Nilai
1	TextBox	Name Text	b 2
2	TextBox	Name Text	c 3
3	CommandButton	Name Caption	Command1 Faktorkan !
4	Label	Name Caption Visible	Hasil1 Hasil1 False
5	Label	Name Caption Visible	Hasil2 Hasil2 False
6	Label	Name Caption Visible	Hasil3 Hasil3 False

Simpan Proyek ini dengan nama: bab062. Selanjunya Double Klik *Command1* (Faktorkan !) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
Dim X1 As Double
Dim X2 As Double
Hasil1.Visible = False
Hasil2.Visible = False
Hasil3.Visible = False K = 1
D = b.Text ^ 2 - 4 * c.Text
E = Abs(D) ^ 0.5
If D < 0 Or (Round(E) - E) <> 0 Then
Hasil1.Visible = True
Hasil1.Caption = "PK tidak dapat difaktorkan"
K = 0
End If
N = Abs(c.Text)
X2 = -N
Do While K = 1 And X2 <= N
X1 = c.Text / X2
If X1 + X2 = b.Text Then
Hasil1.Visible = True
Hasil1.Caption = "<==>(X+" & X1 & "(X + " & X2 & ")=0"
Hasil2.Visible = True
Hasil2.Caption="<==>X+" & X1 & "=0 atau X+" & X2 & "=0"
Hasil3.Visible = True
Hasil3.Caption = "<==>X=" & -X1 & " atau  X=" & -X2
K = 0
End If
If X2 = -1 Then X2 = X2 + 2
```

```
Else
X2 = X2 + 1
End If
Loop
End Sub
```

Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan!
Selanjutnya jalankan program, amati apa yang terjadi!

Latihan

Buat proyek memfaktorkan seperti Sub Bab 6.2, dimana koefisien X2 tidak selalu satu. Buat folder penyimpanan proyek ini dengan nama **Bab_06_03**.

Bab 7
Timer dan Kontrol Program Loop Do Until

Dalam Bab ini kita akan berbicara tentang bagaimana menggunakan bahasa pemrograman dengan kontrol program *Do Until*. Kontrol program ini sama dengan kontrol program *Do While ... Loop* atau *Do ... Loop While*. Jadi mengapa kita harus mempelajarinya? Karena kita mengikuti mata kuliah Pemrograman Komputer maka kita harus mempelajari semua kontrol program yang tersedia. Cara penggunaan kontrol program ini akan dibahas dengan singkat. Selanjutnya kita kita akan membahas penggunaan kontrol/ objek *timer* dalam pembuatan proyek. Setelah selesai mempelajari Bab ini mahasiswa diharapkan mampu menggunakan dan membedakan dua sistimatika kontrol program *Do While Until* dalam penulisan kode Visual Basic serta mampu memanfaatkan kontrol *timer* dalam pembuatan proyek.

7.1. Sintaks Do Until

Loop yang ini mengerjakan perintah antara *Do* sampai dengan *Loop* berulang-ulang sampai syarat/kondisi salah. Sistimatika sintaks adalah :

```
Do Until kondisi
Perintah-Perintah
Loop
```

atau

```
Do
Perintah-Perintah
Loop Until kondisi
```

Contoh 1

Buat folder penyimpanan proyek ini dengan nama **Bab_07_01**. Kita akan membuat proyek penggunaan *Do Until ... Loop* sederhana dengan Form Designer seperti Gambar 5.1 pada Bab 5 dan hanya kode programnya hanya mengganti *While* benar menjadi *Until* salah. Selanjunya Double Klik *CommandButton Command1* (Faktorialnya =) dan lengkapi kode programnya seperti di kolom ke-2 berikut:

Menggunakan While	Menggunkan Until
`Private Sub Command1_Click()`	`Private Sub Command1_Click()`
`N = Angka.Text`	`N = Angka.Text`
`Dim A As Double`	`Dim A As Double`
`Dim i As Integer`	`Dim i As Integer`
`A = 1`	`A = 1`
`i = 1`	`i = 1`

Do **While i <= N**	**Do Until i > N**
A = A * i	A = A * i
i = i + 1	i = i + 1
Loop	Loop
Hasil.Caption = A	Hasil.Caption = A
End Sub	End Sub

Simpan Proyek ini dengan nama: **bab071**. Perhatikan perintah yang tertulis baris demi baris. Apa yang dimaksud dengan perintah tersebut? Jelaskan! Selanjutnya jalankan programnya untuk berbagai nilai, amati apa yang terjadi! Akhirnya berikan kesimpulan!

7.2. Penggunaan Timer dalam Proyek

Pengaturan waktu eksekusi program akan membuat proyek lebih menarik. Bahkan untuk membuat animasi, timer wajib digunakan. Kita akan mempelajari pengaturan waktu eksekusi program dengan dua cara, yaitu: penggunaan loop, dan penggunaan tool timer. Buat folder penyimpanan proyek ini dengan nama **Bab_07_02**. Pertama, kita akan menggunakan loop. Caranya, kode pada Contoh di atas diubah menjadi:

```
Private Sub Command1_Click()
N = Angka.Text
Dim A As Double
Dim i As Integer
A = 1
i = 1
Do Until i > N
A = A * i Print A
i = i + 1
```

```
Loop
Hasil.Caption = A
End Sub
---------------------------------------------
Private Sub tunda()
x = Timer
Do Until Timer > 1 + x
DoEvents
Loop
End Sub
```

Simpan Proyek ini dengan nama: **bab072**, eksekusindan perhatikan hasilnya. Selanjutnya di atas kode *Print A* tuliskan kode *tunggu*, yaitu untuk memanggil fungsi/ prosedur yang kita telah tulis. Kembali eksekusi proyek ini, bandingkan apakah ada yang berbeda dari sebelumnya? Apa yang dimaksud dengan fungsi/ prosedur *tunggu* tersebut? Jelaskan! Berikan pendapatmu!

Contoh 2

Buat folder penyimpanan proyek ini dengan nama **Bab_07_03**. Contoh berikut ini masih sama seperti contoh pada Bab 6, Designer seperti Gambar 6.1, dan hanya ditambah dengan fungsi/prosedur tunggu yang telah kita buat.

Seperti biasa, gunakan beberapa Label untuk membuat persamaan kuadrat dengan dua *TextBox*. Kedua *TextBox* dan *Tool* lain dengan ketentuan sebagai berikut:

No	Tools	Properti	Nilai
1	TextBox	Name Text	b 2
2	TextBox	Name Text	c 3
3	CommandButton	Name Caption	Command1 Faktorkan !
4	Label	Name Caption Visible	Hasil1 Hasil1 False
5	Label	Name Caption Visible	Hasil2 Hasil2 False
6	Label	Name Caption Visible	Hasil3 Hasil3 False

Simpan Proyek ini dengan nama: bab073.

Gambar 6.1 Form Designer Menfaktorkan

Selanjunya Double Klik *Command1* (Faktorkan !) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
Dim X1 As Double
Dim X2 As Double
Hasil1.Visible = False
Hasil2.Visible = False
Hasil3.Visible = False
K = 1
D = Abs(b.Text ^ 2 - 4 * c.Text)
E = D ^ 0.5
If D < 0 Or (Round(E) - E) <> 0 Then
Hasil1.Visible = True
Hasil1.Caption = "PK tidak dapat difaktorkan"
K = 0
End If
N = Abs(c.Text)
X2 = -N
Do While K = 1 And X2 <= N
X1 = c.Text / X2
If X1 + X2 = b.Text Then
Hasil1.Visible = True
Hasil1.Caption="<==>(X+" & X1 & ")(X + " & X2 & ") = 0"
tunggu
Hasil2.Visible = True
Hasil2.Caption="<==>X+" & X1 & "=0 atau      X+" & X2 & "=0"
tunggu
Hasil3.Visible = True
Hasil3.Caption = "<==> X=" & -X1 & " atau X = " & -X2
K = 0
End If
```

```
If X2 = -1 Then
X2 = X2 + 2
Else
X2 = X2 + 1
End If
Loop
End Sub
---------------------------------------------------
Private Sub tunda()
A = Timer
Do Until Timer > 3 + A
DoEvents
Loop
End Sub
```

Selanjutnya jalankan program, amati apa yang terjadi!

Bab ini akan diakhiri dengan cara penggunaan *tool timer*, yang banyak digunakan untuk membuat animasi.

Contoh 3

Buat folder penyimpanan proyek ini dengan nama **Bab_07_04**. Kita akan membuat Proyek dengan design serperti Gambar 7.1. Objek-objek pada Gambar tersebut mempunyai properties seperti berikut:

No	Tools	Properti	Nilai
1	Form	Name Caption	Form1 Timer
2	Label	Name Caption	Label1 SELAMAT DATANG
3	CommandButton	Name Caption	Command1 PLAY
4	CommandButton	Name Caption	Command2 STOP
5	Timer	Name Enabled Interval	Timer1 False 100 (1 = 0,001 detik)

Gambar 7.1. Design Proyek Timer

Klik 2 kali Timer1, lengkapi kode programnya menjadi:

```
Private Sub Timer1_Timer()
Label1.Left = Label1.Left - 10
If Label1.Left < 0 Then
Label1.Left = Form1.Width
End If
```

```
End Sub
```

Klik 2 kali Command PLAY, lengkapi kode programnya menjadi:

```
Private Sub Command1_Click()
Timer1.Enabled = True
End Sub
```

Klik 2 kali Command STOP, lengkapi kode programnya menjadi:

```
Private Sub Command2_Click()
Timer1.Enabled = False
End Sub
```

Run proyek di atas, klik PLAY, klik STOP! Amati apa yang terjadi!

Latihan

Buat proyek yang tulisan pada Labelnya bergerak dengan arah berubah-ubah, misalnya, setelah 10 langkah ke kanan dilanjutkan 10 langkah ke bawah. Setelah itu 10 langkak ke kiri dan selanjutnya 10 langkah ke atas. Buat folder penyimpanan proyek ini dengan nama **Bab_07_04**.

Bab 8
Module dalam Visual Basic

Bab ini akan membahas module pada Visual Basic. Module merupakan tempat kode program tanpa Form. Module merupakan tempat menulis variabel, konstanta, fungsi, dan prosedur. Penggunaan module akan kita terapkan dalam pembuatan proyek pembelajaran sistem persamaan linier. Setelah mempelajari bab ini mahasiswa mampu membuat module dan menggunakannya dalam pembuatan proyek.

8.1. Module

Cara membuat module dalam Project adalah klik Project, Add Module. Sehingga akan tampil jendela Module yang siap ditulis code program. Pada jendela ini, kita dapat mendeklarasikan variabel dengan awalan DIM, PRIVATE, PUBLIC. Kita mendeklarasikan variabel dengan awaln DIM, atau PRIVATE jika variabel tersebut hanya digunakan module yang bersangkutan, tetapi jika variabel akan digunakan program secara keseluruhan maka deklarasi PUBLIC harus menjadi pilihan.

Deklarasi untuk procedure dan fungsi hampir sama dengan variabel, yaitu dimulai dengan DIM, PRIVATE, atau PUBLIC. Prosedur dibuat untuk menangani sejumlah tugas-tugas yang berulang-ulang digunakan. Sedangkan fungsi digunakan untuk mendapatkan variabel terikat dari variabel bebas yang akan digunakan secara berulang-ulang. Sehingga saat dibutuhkan, kita tidak perlu menulis seluruh kode program tetapi cukup hanya memanggil nama prosedur, atau nama fungsinya.

Sistimatika penulisan prosedur:

```
[Private|Public][Static] Sub namaprosedur(argumen)
Perintah-perintah
End Sub
```

Contoh 1

Kode program berikut adalah contoh penulisan program yang digunakan untuk menunda esekusi program yang bernama *tunda*. Eksekusi program akan ditunda selama *i* detik sebagai argumen/ variabel prosedur.

```
Public Sub tunda(i As Integer)
a = Timer
Do Until Timer > i + a
DoEvents
Loop
End Sub
```

Sedangkan sistimatika penulisan fungsi:

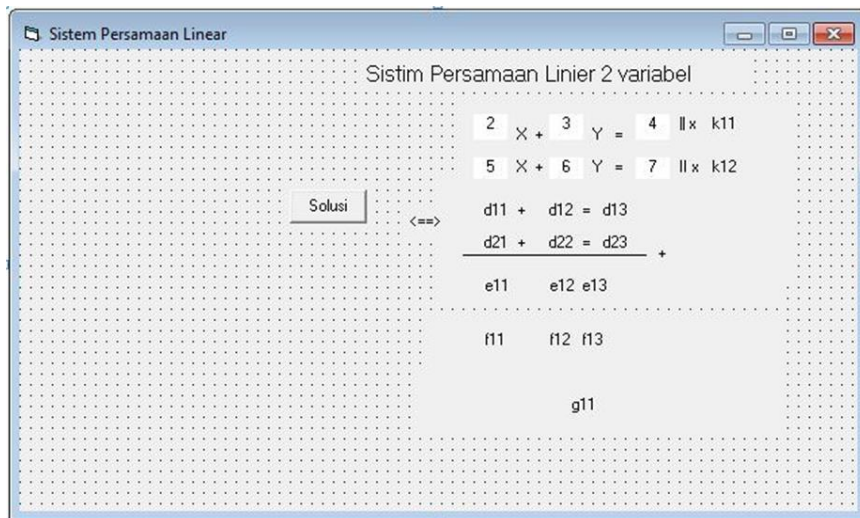
```
[Private|Public][Static]Function namafungsi(argumen)[As type]
Perintah-perintah
namafungsi = variabel output
End Function
```

Contoh 2

Kita akan membuat fungsi menghitung Faktor Persekutuan Terbesar (FPB) dari dua bilangan bulat. Seperti yang kita ketahui FPBnya pasti bernilai bilangan bulat juga.

```
Public Function FPB(a, b) As Integer
ya = 1
If a = 0 Or b = 0 Then
c = 0
ya = 0
End If
If Abs(a) <= Abs(b) Then X1 = Abs(a)
X2 = Abs(b) Else
X1 = Abs(b) X2 = Abs(a)
End If i = X1
Do While ya = 1
If Round(X1/i,0)=(X1/i) And Round(X2/i,0)=(X2/i) Then
c = i
ya = 0
End If
i = i - 1
Loop
FPB = c
End Function
```

8.2. Penggunaan Module dalam Proyek

Buat folder penyimpanan proyek ini dengan nama **Bab_08_01**. Proyek yang akan dibuat adalah menentukan solusi sistim persamaan linier dua variabel. Pada proyek ini kita akan belajar bagaimana membuat module dalam pemrograman visual basic serta cara menggunakannya. Setelah visual basic sudah terbuka kita pertama membuka module dengan klik **Project** >> **Add Module** sehingga akan nampak jendela untuk tempat menulis kode program. Untuk jendela ini, tuliskan kode program prosedur tunda dan fungsi FPB yang ada di sub bab 8.1. Selanjutnya, buat design form seperti Gambar 8.1.

Gambar 8.1. Design Sistem Persamaan Linear

Properties tool *Comand1* Solusi dan sistem persamaan linear dua variabel pertama akan dijelaskan pada tabel berikut. Sedangkan nama properties selanjutnya akan ditulis sesuai dengan nama *Caption*-nya dan *False* nilai *visible*-nya.

No	Tools	Properti	Nilai
1	CommandButton	Name Text	Command1 Solusi
2	TextBox	Name Text	a11 2
3	TextBox	Name Text	b11 3
4	TextBox	Name Text	c11 4
5	TextBox	Name Text	a12 5
6	TextBox	Name Text	b12 6
7	TextBox	Name Text	c12 7

Selain itu, caption dengan simbol + dan = mempunyai nama khusus dan mohon tebak namanya setelah melihat kode program berikutnya. Selain itu kita masih memasukkan Label nama eki1 dengan simbol ekivalen, dan nama L1 untuk garis. Simpan Proyek ini dengan nama: **bab081**. Selanjunya Double Klik *Command1* (Solusi) dan lengkapi kode programnya seperti berikut:

```
Private Sub Command1_Click()
Dim v111, v112, v121, v122 As Integer
```

```
Form1.Refresh
d11.Visible = False
d12.Visible = False
d13.Visible = False
d21.Visible = False
d22.Visible = False
d23.Visible = False

'nilai-nilai kefisien
v011 = a11.Text
v012 = b11.Text
v013 = c11.Text
v021 = a12.Text
v022 = b12.Text
v023 = c12.Text

c1 = FPB(v011, v021)
c2 = FPB(v012, v022)
Form1.ForeColor = vbRed
Print "FPB koefisien X: " & c1 & ", koefisien Y: " & c2
Print ""
tunda (1)
If c1 >= c2 Then
Print "Kita akan menyamakan koefisien X"
pk11 = Abs(a12) / c1
pk12 = Abs(a11) / c1
k = 1
Else
Print "Kita akan menyamakan koefisien Y"
pk11 = Abs(b12) / c2
pk12 = Abs(b11) / c2
k = 2
```

```
End If
Print "Sehingga"
Print " persamaan (1) dikali " & pk11
Print " persamaan (2) dikali " & pk12
tunda (2)
Print " yaitu lihat ===>>"
tunda (1)
p11.Visible = True
k11.Visible = True
k11.Caption = pk11
tunda (1)
p12.Visible = True
k12.Visible = True
k12.Caption = pk12
eki1.Visible = True

'Nilai koefisien hasil perkalian
v111 = pk11 * v011
v112 = pk11 * v012
v113 = pk11 * v013
v121 = pk12 * v021
v122 = pk12 * v022
v123 = pk12 * v023

'Memilih ditambah atau dikurang
If k = 1 Then
If (v111 * v121) < 0 Then t = 1
Else
t = -1
End If
Else
If (v112 * v122) < 0 Then
```

```
t = 1
Else
t = -1
End If
End If

L1.Visible = True
O1.Visible = True
If t = 1 Then
O1.Caption = "+"
Else
O1.Caption = "--"
End If

d11.Visible = True
d11.Caption = v111 & " X"
T11.Visible = True

If v112 >= 0 Then
T11.Caption = "+"
Else
T11.Caption = "--"
End If
d12.Visible = True
d12.Caption = Abs(v112) & " Y"
s11.Visible = True
d13.Visible = True
d13.Caption = v113

d21.Visible = True
d21.Caption = v121 & " X"
T12.Visible = True
```

```
If v122 >= 0 Then
T12.Caption = "+"
Else
T12.Caption = "--"
End If
d22.Visible = True
d22.Caption = Abs(v122) & " Y"
s12.Visible = True
d23.Visible = True
d23.Caption = v123
v213 = v113 + t * v123

e13.Visible = True
f13.Visible = True
g11.Visible = True
If k = 1 Then
e12.Visible = True
v212 = v112 + t * v122
e12.Caption = v212 & " Y"
e13.Caption = "= " & v213
f12.Visible = True
f12.Caption = "   Y"
f13.Caption = "= " & v213 & "/" & v212
g11.Caption = "Y = " & v213 / v212
Else
e11.Visible = True
v211 = v111 + t * v121
e11.Caption = v211 & " X"
e13.Caption = "= " & v213
f11.Visible = True
f11.Caption = "   X"
f13.Caption = "= " & v213 & "/" & v211
```

```
g11.Caption = "X = " & v213 / v211
End If
End Sub
```

Selanjutnya jalankan program, amati apa yang terjadi!

Latihan

1. Kode program contoh proyek di atas belum sempurna. Sempurnakan kode program tersebut!
2. Buat proyek untuk menentukan sistim persamaan linear tiga variabel. Buat folder penyimpanan proyek ini dengan nama **Bab_08_02.**

Bab 9
Database

Database (basis data) dapat diartikan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data, seperti menambah dan menghapus data. Bab ini merupakan pengenalan bagaiamana cara membuat sistim informasi manajemen pengelolaan pendidikan. Kita akan membahas struktur database, membuat database, membuat tabel, dan membuat laporan. Setelah mempelajari bab ini mahasiswa mampu membuat sistem informasi manajemen pendidikan yang sederhana. Buat folder penyimpanan proyek ini dengan nama **Bab_09_01**. Semua proyek dan file akan disimpan dalam folder tersebut.

9.1. Struktur Database

Struktur database adalah cara merencanakan hubungan antara semua tabel yang akan digunakan dalam pembuatan database. Sebelum strukturnya, kita terlibih dahulu menuliskan tabel yang

diperlukan dan data apa yang diperlukan untuk setiap tabel. Setiap tabel harus mempunyai kolom kunci (*Primery Key*).

Untuk sistem informasi manajemen pendidikan yang akan dibuat dalam modul ini, kita membutukkan beberapa tabel, antara lain: Tabel Identitas Mahasiswa (IM), Tabel Dosen Pengajar (DP), Tabel Mata Kuliah (MK), Tabel Kartu Rencana Studi (KRS). Setiap tabel mempunyai field/kolom yang berisi kumpulan data-data yang sejenis. Selengkapnya tabel-tabel di atas akan ditulis seperti berikut:

Tabel IM

No.	Nama Field	Tipe Data	Ukuran
1	NPM	Text	8
2	NamaMahasiswa	Text	30
3	Prodi	Text	30
4	Alamat	Text	40

Primery Key adalah NPM

Tabel DP

No.	Nama Field	Tipe Data	Ukuran
1	NIDN	Text	11
2	NamaDosen	Text	30
3	Prodi	Text	30
4	Alamat	Text	40

Primery Key adalah NIDN

Tabel MK

No.	Nama Field	Tipe Data	Ukuran
1	KodeMK	Text	8
2	NamaMK	Text	30

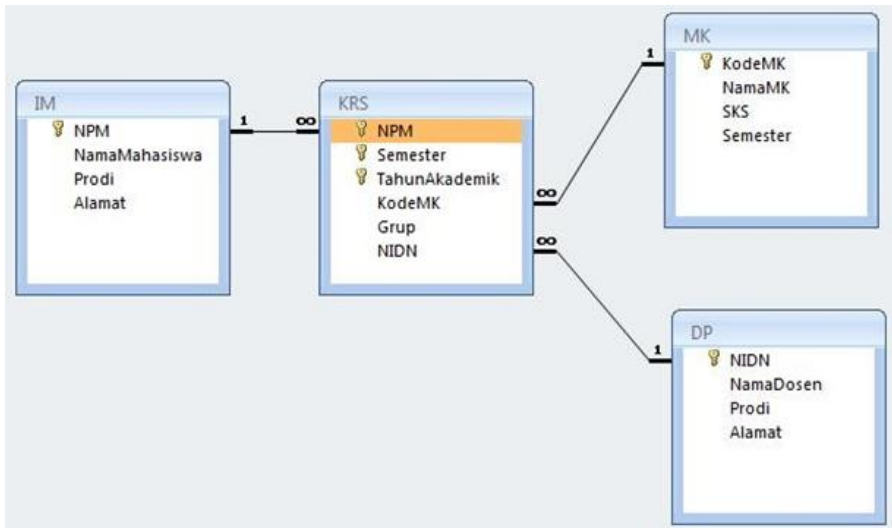
3	SKS	Number	Integer
4	Semester	Text	6

Primery Key adalah KodeMK

Tabel KRS

No.	Nama Field	Tipe Data	Ukuran
1	NPM	Text	8
2	Semester	Text	6
3	TahunAkademik	Text	9
4	KodeMK	Text	8
5	Grup	Text	1
6	NIDN	Text	11

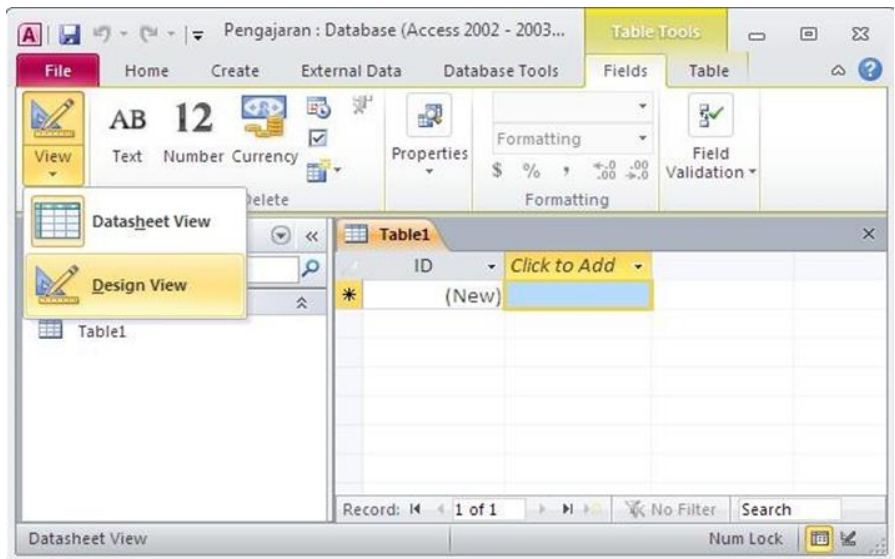
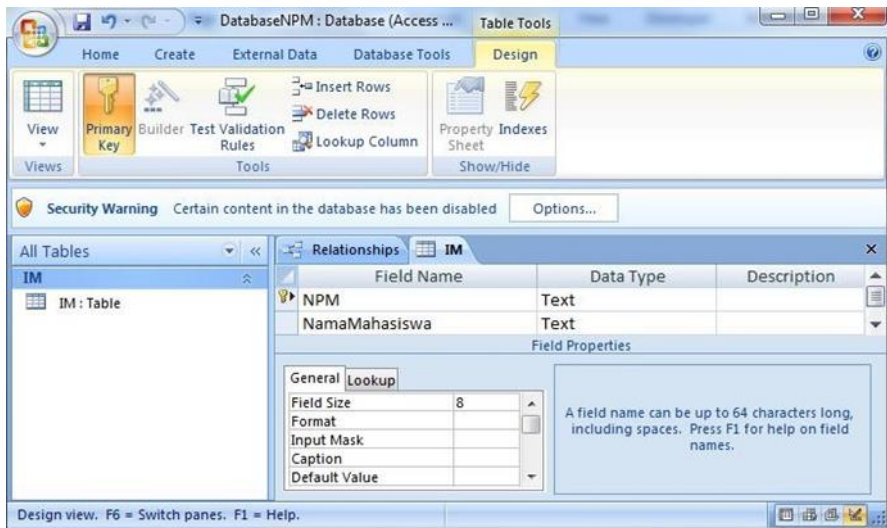
Primery Key adalah NPM, Semester, TahunAkademik

Relasinya adalah sebagai berikut:

9.2. Membuat Database

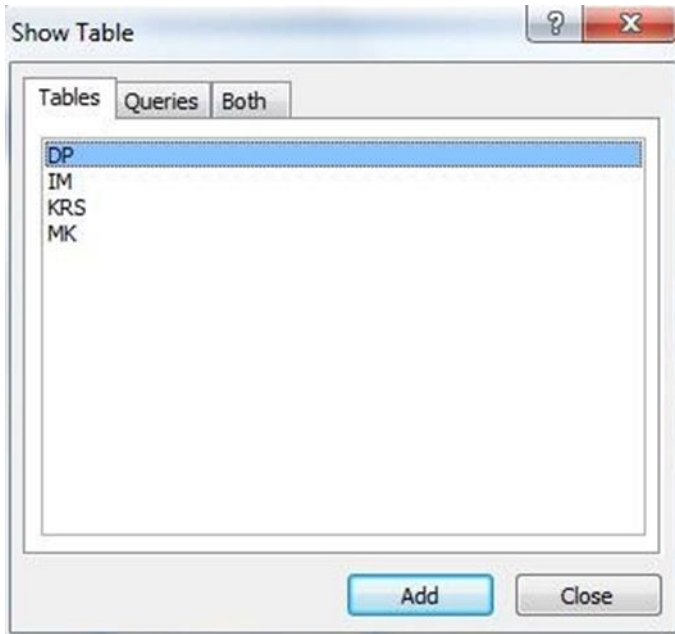
Database Management System (DBMS) merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan database. Salah satu macam DBMS yang populer dewasa ini berupa RDBMS (*Relational DataBase Management System*), yang menggunakan model basis data relasional atau dalam bentuk tabel-tabel yang saling terhubungkan. **Microsoft Access dan MySQL** merupakan contoh **produk RDBMS**. Dalam mata kuliah ini, kita akan menggunakan Microsoft Access untuk pengelolaan database. Langkah pembuatannya sebagai berikut:

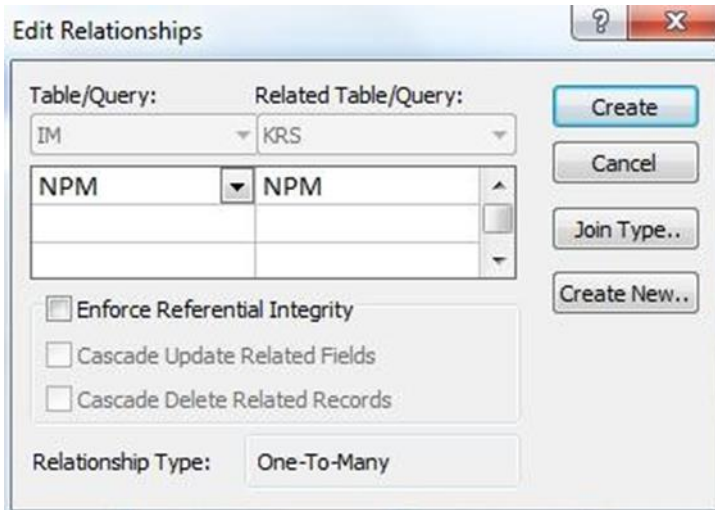
1. Jalankan program Microsoft Acces (**Start >> All Programs >> Microsoft Office >>Microsoft Office Access**).
2. Klik **Blank Database**.
3. Buat **DatabaseNPM** sebagai nama databasenya, pilih format file *.mdb (format MS Access 2000), dan simpan di folder yang sesuai.

Selanjutnya kita akan membuat tabel-tabel yang dibutuhkan. Pada saat ini kita akan membuat Tabel IM dan tabel-tabel lain sebagai latihan. Cara membuatnya adalah sebagai berikut:

1. Klik **Create table in Design View.**

2. Isikan Field Name dengan nama field dari Tabel IM. Untuk Data Type (Tipe Data) dapat diisikan dengan memilih dari daftar yang telah ada. Untuk mengatur ukuran field dapat dilakukan dengan mengganti properti Field Size (pada tab General) dari masing-masing field.

3. Klik di sebelah kiri NPM untuk memilih baris ini, dan selanjutnya klik *Primary Key*.
4. Simpan tabel dengan nama IM.

Lakukan hal yang sama untuk membuat dan mengisi data pada Tabel **DP**, Tabel **MK**, dan Tabel **KRS**.

Setelah selesai semua tabel dibuat dan disimpan, klik **Database Tools >> Relationships** sehingga tampil seperti berikut:

Blok semua tabel dan klik **Add**. Setelah itu kita akan menghubungkan nama-nama field yang bersesuaian seperti pada relasi yang ada pada akhir Sub Bab 9.1. Cara untuk menghubungkan **NPM** pada Tabel **IM** dan Tabel **KRS** adalah:

1. Klik dan tahan NPM pada tabel IM, dan sambil tarik sampai NPM pada tabel KRS. Setelah itu lepaskan, akan tampak seperti berikut:

2. Centang **Enforce Referential Integrity**
3. Klik **Create**

Lakukan hal yang sama untuk relasi yang lain! Jangan lupa menyimpannya!

9.3. Membuat Form

Pada sub bab ini kita akan membuat form yang mempunyai kemampuan Multi Document Interface (MDI), yaitu form yang mempunyai kemampuan menampilkan beberapa form sekaligus. Jika kita ingin membuat proyek yang mempunyai menu maka kita harus memnggunakan form MDI. Caranya:

1. Masuk ke proyek visual basic.

2. Karena form yang ditampilkan bukan MDI, maka klik kanan **Form1** yang ada di jendela Project Explorer, selanjutnya klik **Remove Form1**.
3. Klik kanan **Project1** pada jendela Project Explorer, dan klik **Add >> MDI Form**.
4. Klik **Open**.

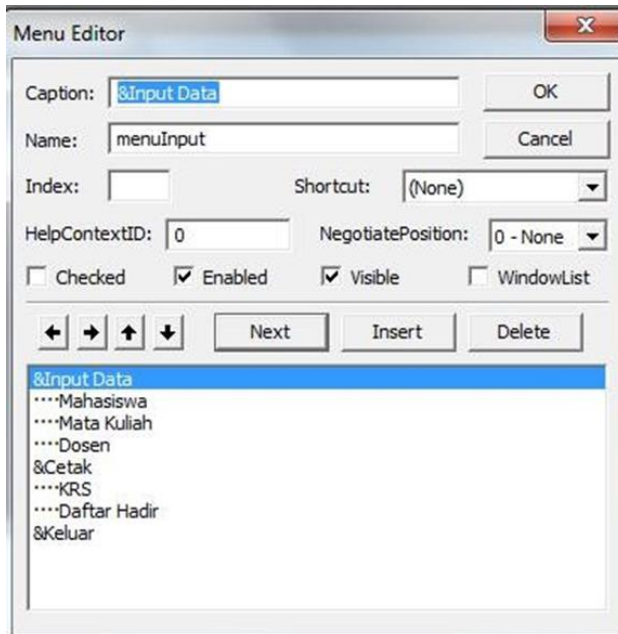
Maka Form yang terbentuk adalah Form MDI. Selanjutnya atur propertiesnya menjadi:

No	Tools	Properti	Nilai
1	MDIForm	Nama Caption WindowState	frmMDI Data Akademik 2-Maximized

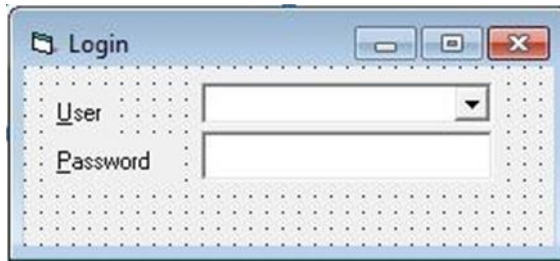
Selanjutnya kita akan meletakkan menu dengan struktur:

&Input Data	&Cetak	&Keluar
Mahasiswa	KRS	
Mata Kuliah	Daftar Hadir	
Dosen		

Caranya:

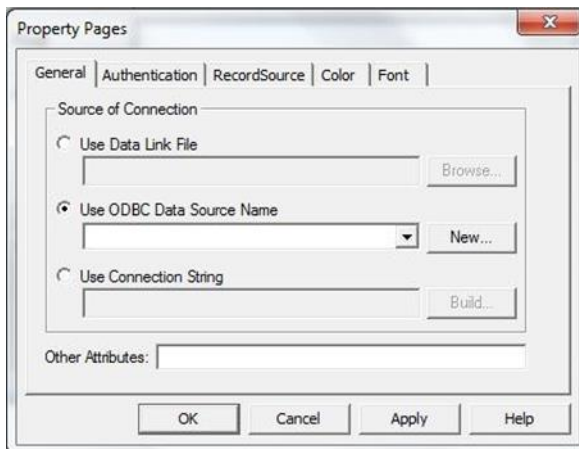
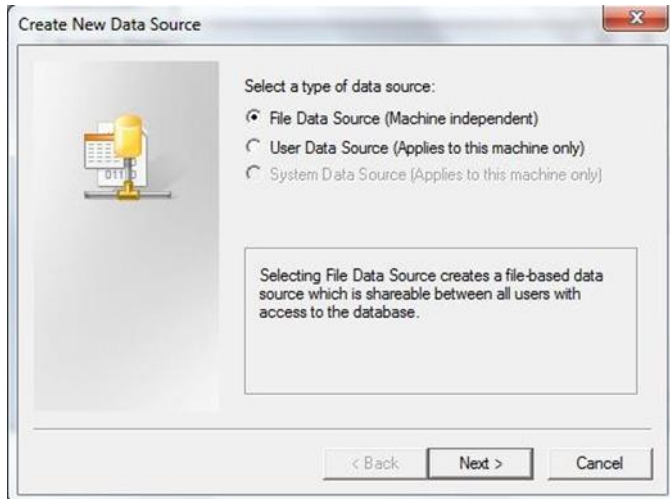
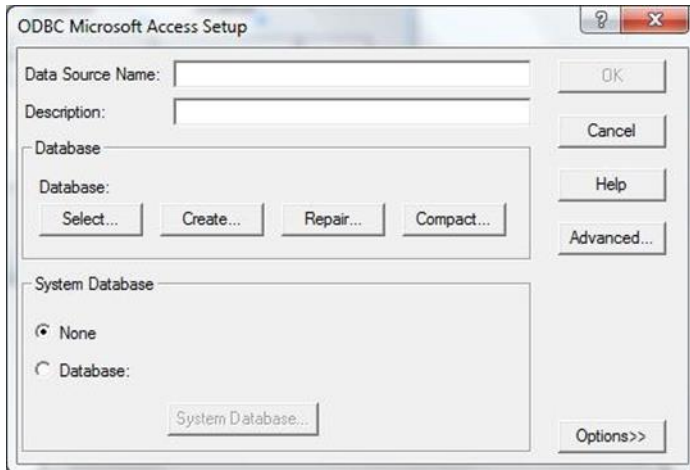
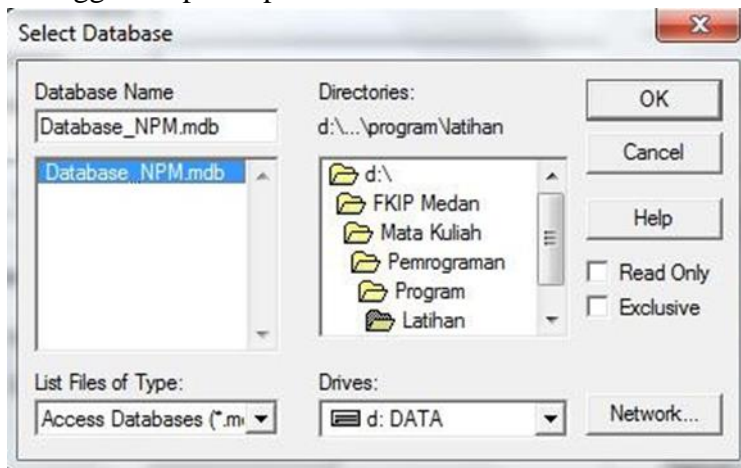
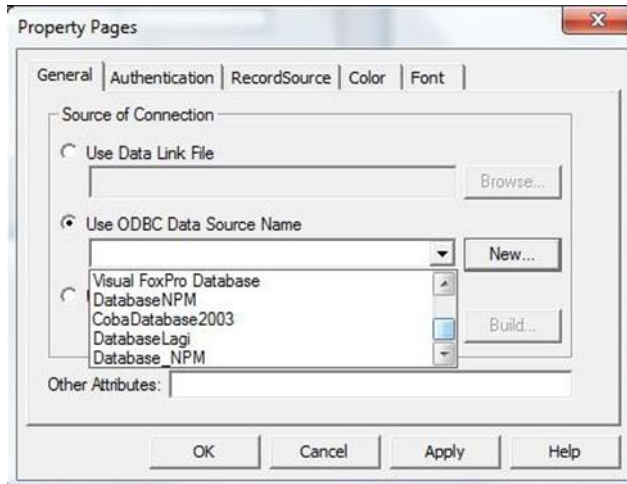
1. Klik **Menu** Editor dari Toolbar Standard.
2. Isi semuanya sampai kelihatan seperti gambar beikut:

3. Klik OK

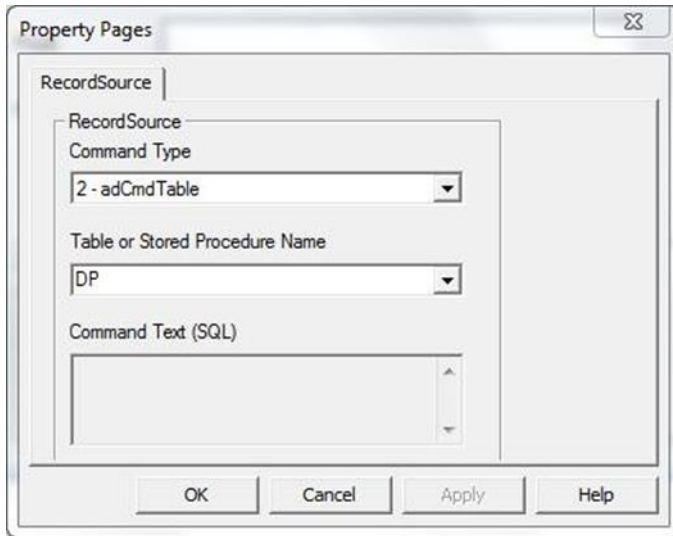
Selanjutnya kita ingin membatasi orang yang dapat mengakses proyek ini dengan cara membentuk form login, caranya:

1. Klik kanan **Form MDI** pada jendela Project Explorer, pilih **Add >> Form**
2. Pada kotak dialog, klik **Form** >> **Open**
3. Sehingga akan bertambah satu form kosong dan design menjadi berikut:

4. Lengkapi propertiesnya menjadi:

No	Tools	Properti	Nilai
1	Form	Name Caption MDIChild WindowState	frmDP Input Data Dosen True 2-Maximized
2	TextBox	Name Text	dpNIDN
3	TextBox	Name Text	dpNamaDosen
4	TextBox	Name Text	dpProdi
5	TextBox	Name Text	dpAlamat
6	Command	Nama Caption	dpTambah Tambah
7	Command	Nama Caption	dpUbah Ubah
8	Command	Nama Caption	dpHapus Hapus

9	Adodc	Nama Caption	Adodc1 Data Dosen
10	DataGrid	Nama	DataGrid1

5. Ada kemungkinan tools **Adodc** dan **DataGrid** belum tersedia pada ToolBox, caranya klik kanan pada jendela ToolBox pilih **Components**, selanjutnya centang kotak **Microsoft ADO Data Control 6.0 (OLEDB)**, dan kotak **Microsoft DataGrid Control 6.0 (OLEDB)**, dan akhiri dengan klik **OK**. Masukkan ke form untuk melengkapi design.
6. Selanjutnya klik kanan **Adodc1 >> ADODC Properties** dan muncul

7. Pilih **Use ODBC Data Source Name**, klik **New** dan akan tampak

8. Pilih **User Data Source**, klik **Next**, **pilih Microsoft Access Driver** (***.mdb**,***.accdb**), klik **Next**. Klik Browse untuk mencari folder penyimpanan file database yang telah dibuat dengan Microsoft Access. Setelah file databasenya sudah ditemukan klik **Save**, klik **Next**, dan akhirnya klik Finish. Kembali ke jendela sebelumnya yaitu:

Klik **Select** dan lanjutkan dengan mencari nama **file Database(NPM)** pada folder tempat menyimpan database, sehingga nampak seperti berikut:

Klik **OK**, dan selanjutnya tuliskan pada Data Source Name: **Databse_NPM** (sesuai nama database yang anda buat), dan klik **OK**. Kembali ke jendela sebelumnya

Pilih **Database_NPM** pada Use ODBC Data Source Name. Klik **Apply >> OK**.

9. Ubah properties **Adodc1** untuk nilai dari RecordSource dengan cara klik (...) dan akan muncul jendela Property Pages dan lengkapi seperti berikut

10. Klik **OK**. Periksanilai properties **ConnectionString** adalah **DSN=DatabaseNPM** sesuai dengan nama database.
11. Pilih tools **DataGrid**, dan ubah nilai properties **DataSource** menjadi **Adodc1**.
12. Lengkapi code program untuk klik command **dpTambah**, **dpUbah**, **dpHapus**, dan **klik DataGrid1** seperti berikut:

```
Private Sub DataGrid1_Click()
dpNIDN.Text = Adodc1.Recordset!NIDN
dpNamaDosen.Text = Adodc1.Recordset!NamaDosen
dpProdi.Text = Adodc1.Recordset!Prodi
dpAlamat.Text = Adodc1.Recordset!Alamat
End Sub

Private Sub dpHapus_Click()
```

```
Adodc1.Recordset.Delete
dpNIDN.Text = ""
dpNamaDosen.Text = ""
dpProdi.Text = ""
dpAlamat.Text = ""
End Sub


Private Sub dpTambah_Click()
Adodc1.Recordset.AddNew
Adodc1.Recordset!NIDN = dpNIDN.Text
Adodc1.Recordset!NamaDosen
dpNamaDosen.TextAdodc1.Recordset!Prodi=dpProdi.Text
Adodc1.Recordset!Alamat = dpAlamat.Text
Adodc1.Recordset.Update
End Sub

Private Sub dpUbah_Click()
Adodc1.Recordset!NIDN = dpNIDN.Text
Adodc1.Recordset!NamaDosen = dpNamaDosen.Text
Adodc1.Recordset!Prodi = dpProdi.Text
Adodc1.Recordset!Alamat = dpAlamat.Text
Adodc1.Recordset.Update
End Sub
```

13. Simpan.

Kita akan menghubungkan form frmDP dengan menu yang ada pada Form MDI dengan cara:

1. Pilih form MDI yang sudah dibuat, klik Input **Data** >> **Dosen**.
2. Lengkapi kode programnya menjadi:

```
Private Sub menuDosen_Click()
frmDP.Show
End Sub
```

3. Run, klik **Input Data >> Dosen** dan akan tampil seperti gambar berikut:

4. Lengkapi data Tabel DP dengan mengetik langsung pada TextBox serta memanfaatkan tombol Tambah, Ubah, Hapus

NIDN	Nama Dosen	Prodi	Alamat
0028086808	Efron Manik	P. Matematika	Marindal
0004087104	Simon Panjaitan	P. Matematika	Simpang Limun
0110117401	Sanggam P. Gultom	P. Matematika	Padang Bulan
0102028403	Lena R. Pangaribuan	P. Matematika	Polonia

Dengan cara yang sama tolong buat Form untuk Input Data Identitas Mahasiswa (IM).

NPM	NamaMahasiswa	Prodi	Alamat
15150206	Elviana Tampubolon	P. Matematika	Marindal
15150207	Masrina Sinaga	P. Matematika	Simpang Limun
15150209	Boy Simatupang	P. Matematika	Padang Bulan
15150210	Masdawaty Sitohang	P. Matematika	Polonia
15150211	Wahyu Prima Pasaribu	P. Matematika	Pancing

Dengan cara yang sama tolong buat Form untuk Input Data Mata Kuliah (MK).

KodeMK	NamaMK	SKS	Semester
15150206	Etika	2	2
15150207	Statistika Dasar	3	5
15150209	Program Linier	3	6

Latihan

1. Kode program contoh proyek di atas belum sempurna. Sempurnakan kode program tersebut!
2. Buat proyek untuk menetukan sistim persamaan linear tiga variabel.

Daftar Pustaka

Boehm, Anne (2008). *Murach's Visual Basic 2008 Mike Murach Series: Training and Reference Series*. Mike Murach & Associates.

Liberty, Jesse (2005). *Programming Visual Basic 2005 O'Reilly Series Safari Books Online*. O'Reilly Media, Inc.

Liberty, Jesse (2003). *Programming Visual Basic .Net O'Reilly Series Safari Books Online*. O'Reilly Media, Inc.

Petroutsos, Evangelos (2010). *Mastering Microsoft Visual Basic 2010*. John Wiley and Sons.

Phil, Jones (2001). *Visual Basic: A Complete Course Letts Higher Education List Series*. Cengage Learning EMEA.

Visual Basic 6.0
untuk Media Pembelajaran Interaktif

Visual Basic adalah bahasa pemrograman berorientasi objek sehingga mudah dipelajari dan langsung dapat disimpan dalam file aplikasi. Pembahasan buku ini dimulai dalam bentuk yang sederhana dengan dilengkapi banyak contoh-contoh sehingga pemula juga dapat belajar sendiri untuk mempelajari buku ini. Kode program yang digunakan adalah kode yang ditulis dalam Bahasa Visual Basic 6.0. Sehingga sebaiknya belajar sambil praktik merupakan cara terbaik dalam mempelajari buku ini.

Buku ini dimulai dengan penjelasan tentang diagram alur dan algoritma. Hal ini sangat dibutuhkan dalam pembuatan proyek dengan kode program yang sangat panjang. Selanjutnya jenis-jenis tools akan diperkenalkan berserta cara penggunaannya. Kontrol program *if*, *case*, *for next*, *do while*, dan *do until* akan dipelajari dengan cara penggunaannya. Buku ini diakhiri dengan pembuatan database dalam Visual Basic 6.0. Contoh-contoh proyek dalam buku ini berbentuk media pembelajaran interaktik, sehingga buku ini berguna untuk pembutan media pembelajaran di era industry 4.0

ISBN 978-623-95324-0-6

9 786239 532406

Penerbit:
LPPM UHN Press